Speaker Recognition using Orthogonal LPC Parameters in Noisy Environment

by

Asad Kamal Abbasi

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

SYSTEMS ENGINEERING

January, 1985

Order Number 1355685

Speaker recognition using orthogonal LPC parameters in noisy environment

Abbasi, Asad Kamal, M.S.

King Fahd University of Petroleum and Minerals (Saudi Arabia), 1985

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

University of Petroleum and Minerals

SPEAKER RECOGNITION USING
ORTHOGONAL LPC PARAMETERS IN NOISY
ENVIRONMENT

BY

ASAD KAMAL ABBASI

A Thesis Presented to the
FACULTY OF THE COLLEGE OF GRADUATE STUDIES

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

IN

SYSTEMS ENGINEERING

JANUARY 1985

Printed By: UPM Printing Press

UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

COLLEGE OF GRADUATE STUDIES

This thesis, written by

ASAD KAMAL ABBASI

under the direction of his Thesis Committee, and approved by all its members, has been presented to and accepted by the Dean, College of Graduate Studies, in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE
IN
SYSTEMS ENGINEERING

Dean, College of Graduate Studies

Date: Dec. 16, 84

Deapartment Chairman

Thesis Committee

Chairman

S. Selim Dec 4, 84
Member

Member

Dedicated to

My Beloved Wife Neelo,
Dear Sons Taemoor & Humayoon
And
Loving Parents.

ACKNOWLEDGEMENTS

A number of persons helped me on this work. I thank them all. The bibliography identifies the scholars whose work on this field I used or referred to.

I am highly indebted to my thesis advisor Dr. M. Shahgir Ahmed who provided me with all the help he could. His suggestions have been indispensible and I am very much thankful to him.

Drs. S. Z. Selim and N. E. Gough have been helpful in the preparation of this final version by proposing a number of corrections. Thanks to them.

I am also thankful to my friends whose speech have been used in the experiments.

Special thanks are due to my very good friend Tahir Mufti who helped me with his very valuable suggestions.

Last, but in no way the least, I am extremely grateful to my family, specially my mother, wife and sons for their moral support and patience.

ABSTRACT

Last two decades witnessed a number of speaker recognition algorithms that yield reasonably good to excellent performance with high quality and relatively noise-free speech. However, in practice, noise is unavoidable. The presence of noise is found to cause severe deterioration in the performance of these algorithms.

In this thesis, the problem of recognizing speaker from noisy speech has been studied. Sambur's algorithm that utilizes orthogonal linear predictive coding (OLPC) parameters is chosen for this study because of its simplicity and high recognition accuracy. The effect of noise on the performance of Sambur's algorithm is studied. The algorithm is, then, modified by incorporating various parameter estimation techniques which are capable of yielding relatively better LPC parameters. These techniques may be divided into two groups. The first group contains three parameter estimation procedures, namely, the instrumental variable (IV) method, the autocorrelation subtraction (AS) method and estimation through the use of shifted Yule-Walker (SYW) equations.

The techniques in the second group are based upon enhancing speech followed by the conventional LPC estimation technique applied to the enhanced speech. Three enhancement algorithms have been tested, namely, the adaptive noise cancellation (ANC) technique, the linear predictive smoothing (LPS) technique and the adaptive filtering technique (AFT).

It was found from comparative study that speaker recognition based on speech enhanced by AFT yielded the best recognition accuracy among these techniques.

بسم الله الرحمن الرحيم

ملخص البحث

شهد العقدان الأخيران عدة طرق للتعرف الآلى على المتحدث وهى تعطى نتائج تتراوح ما بين المعقول الى الممتاز فى حالة الحديث الخالى من التشويش وذو الجوده العاليه . ولكن تفادى التشويش غير ممكن فى الواقع.

ولقد وجد أن التشويش يسبب انحطاط بالغ فى كفاءة هذه الطرق ، وفى هذه الرساله ندرس مشكلة التعرف على المتحدث من خلال حديث مشوش ، وقد اخترنا طريقة سامبر التى تستعمل بارامترات متعامدة خطيه تنبئويه فى هذه الرساله لبساطتها ودقتها فى التمييز .

ودرسنا تأثير التشويش على اداء طريقة سامبر ، ثم عدلنا فى الطريقه باضافة طرق مختلفه لايجاد البرامترات للحصول عليها بدقه افضل.

ويمكن تقيسم هذه الطرق الى مجموعين :-

المجموعه الاولى : تحتوى على ثلاثة طرائق لتقدير البرامترات وهى طريقه المتغير الفعال وطريقة طرح معاملات الارتباط وطريقه معادلات يول - ووكرز .

اما طرق المجموعه الثانيه : فهى تعتمد على تقوية الحديث ثم استخدام الطريقه المعتاده للبرامترات المتعامده الخطيه التنبئويه على الحديث المقوى .
ولقد استخدمت ثلاثة طرق للتقويه وهى طريقة حذف التشويش التطويعيه وطريقة التنعيم الخطى وطريقة التصفيه .

ولقد وجد من خلال دراسة مقارنه ان التعرف على المتحدث بعد تقويه الحديث بطريقة التصفيه هى افضل من باقى الطرق من ناحية دقة التمييز.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter I

INTRODUCTION

1.1 SPEAKER RECOGNITION

During the past twenty years there has been an increasing interest in the identification of individuals by automatic analysis of samples from their speech. The general problem has a number of variations due to a large number of forseeable application areas. The application of greatest commercial interest seems to be in confirming the identity of persons carrying business transactions, especially in cases where other means of identification are unavailable or are not appropriate. An example is business carried out by telephone.

Another useful application is in entry control. Lock and key have been the traditional means for this purpose. A great improvement in the flexibility and usefulness of entry control systems has recently been achieved with the help of badges, secret numbers and automatic control. However, virtually all entry control systems in use today share a serious shortcoming: verification of identity is based upon possession of an artifact and is thus vulnerable. The ultimate solution is to verify a person's identity based upon a *personal attribute* unique to him.

A person's speech is an appropriate attribute for identification and is well suited to computer processing. The use of speech also has other advantages, including the need of an inexpensive transducer (microphone), fast verification with almost fully automatic procedures, and a flexible medium for controlling verification performance. The inherent dynamicity of speech signal is, however, the primary limitation in using speech. Speech characteristics are functions of time and cooperation of the speaker is required to minimize the variability of the speech parameters.

As a third example consider the case of crime investigations where authorities are often facing the problem of identifying persons by comparing recorded voices. Such situations arise in instances of kidnapping and bomb threats etc. Although the human listener's capability of recognizing voices is considerably high, human performance is restricted to a value that is below a margin to be used for evidence at law courts. This is because of the extreme high range of variations that is inherent to the human speech.

In the subject of speaker recognition, generally, two types of problems are considered: i)verification of a claimed identity called *the speaker verification*, and ii)identification of an unknown speaker called *the speaker identification*. In most of the cases where the speaker identifies himself, the problem reduces to verification only. In other words the system is required to verify the identity rather than identifying a person.

It is of importance to realize the difference between speaker identification and speaker verification. From statistical point of view, the probabilities of making a correct decision are very much different in the two cases. In case of identification the total number of decisions is the same as the number of speakers to be identified. Hence the probability of making an error increases with the population size.

Theoretically speaking, the probability of a correct decision approaches zero as the number of speakers is sufficiently increased. Verification, on the other hand, consists of the matching of only two speech samples resulting in a single decision with two possible values, i.e. either the claimed identity is accepted or rejected. Hence the expected probability of error is comparatively very low and is independent of the population size.

Speaker recognition becomes simpler if all the speakers are required to utter the same utterence. This also results in a better recognition accuracy. This restriction divides the basic problem into two categories, namely, speech dependent and speech independent recognition. Let us discuss the above mentioned concepts in some detail.

Any speaker recognition problem starts with a defined population of speakers which we will call *the customer set*. The person who is to be identified will be called *the test speaker*. The speech templates of all the speakers in the customer set are formed by extraction of a number of parameters from their speech samples. These parameters can be chosen from a large set of possibilities like formant frequencies, formant amplitudes, LPCs, PARCOR coefficients, log area coefficients and many others. They can be extracted either directly from speech data or after a spectral transformation to the frequency domain. The choice of a set of parameters depends upon the particular recognition procedure adopted. Hypothetically, these parameters can be divided into two categories, i)those which bear speech information and ii)those which bear information about the speaker. This gives rise to one of the most important steps towards successful speaker recognition, that is, the selection of those parameters that efficiently represent the speaker dependent information in speech. Procedures for the selection of these parameters can be found in [1-8]. An ideal set of recognition parameters should exhibit at least the following characteristics [1] :-

1. high efficiency in representing speaker dependent information,
2. easy to measure,
3. stable over time,
4. frequent and natural occurence in speech,
5. insensitivity to speaking environment, and
6. insusceptibility to mimicry.

Once the parameters have been extracted, the next step is essentially the comparison of the speech parameters of the test speaker with those of the individuals in the customer set. This involves the calculation of the distances between points in the multi-dimensional speech parameter space. Different norms have been used as distance measures including Euclidean metric, Mehalanobis distance, log-likelihood ratio statistic [9], and others [4,10,11].

Recognition error rate is the key parameter in adequately describing or specifying the performance of a speaker recognition system. Other factors responsible for the user acceptance are total recognition transaction time and overall system performance. Recognition errors may be categorized as Type I and Type II. A Type I error refers to the situation where a claimed identity is rejected when in fact the claim was true. A Type II error is the acceptance of the claimed identity when in fact the identity claim was false. Recognition errors may occur due to the following three reasons:

1. Variations in speech by the same speaker,
2. Similarities in speech between speakers, and
3. External influences like problems in automatic measurement, comparison and variable recording environment.

Variations by the same speaker result in Type I errors. Abnormal variations in speech include laryngitis and the common cold. These variations often cause user rejection. Occasionally, however, normal variations like those in rate of speech, voice quality and articulation may also cause Type I error.

Similarities between speeches by different speakers are the source of Type II errors. In addition to an occasional natural similarity, Type II errors occur as a result of artful mimicry or through the use of tape recording. Appropriate choice of speech parameters can minimize the probability of successful mimicry. The use of a tape recording can be defeated by randomized choice of speech data.

External influences include variations in recording environment, errors in time alignment of test and reference speech (if such a scheme is used), errors in feature extraction algorithms, and inadequate representation of speech characteristics. Variations in speech signal environment, in turn, include differences between microphones, mouth-to-microphone distance, difference in azimuth and most of all addition of noise in the test speech. It may be shown [12] that the expected probability of error for speaker verification is independent of the customer set size, while in case of speaker identification as the customer set size increases the problem becomes more difficult and the probability of error approaches 1.

For the purpose of speaker verification, depending upon the population and the particular distance measure chosen, a threshold value of the distance measure is decided. The claimed identity is verified if the distance between the parameter set of the test speaker and that of the claimed customer is less than the threshold. This concept has been stated by Doddington [12] as: the claimed identity is accepted if the likelihood of the data given his claimed identity is large enough when the test speaker is tried against a speaker chosen at random from the customer population. In the strict sense the probability of error will be a function of the test speaker, the claimed identity and the customer population. However it can be safely assumed that the expected probability of error with respect to each individual in the customer set is independent of the population size. It is also assumed that the probability of the claimed identity being true is independent of the population size. As for the effort involved, it is clear that, since in speaker verification only a single comparison of parameters is required to verify the claim, the effort is independent of the population size.

Next, let us consider speaker identification for a customer set of size N and a test speaker. The problem can be stated as, 'given that the test speaker is in the customer set, find that individual in the set who is most likely to be the test speaker'. In this case the distance of the speaker recognition parameters is calculated from those of each individual in the customer set. The test speaker is, then, identified to be the one corresponding to the minimum distance. In other words, that speaker in the customer set is identified as the test speaker who has the greatest likelihood of generating the recognition parameters corresponding to the test speaker. The probability of error when comparing the test speaker with an individual in the customer set will be a function of the two speakers. But as in the previous case it can be assumed that the expected probability of error obtained by averaging the effects of the comparison over each individual in the customer set will be invariant of the population size. However, as in the identification process, N comparisons are involved, the probability of incorrect decision increases as N increases. Also it is quite obvious that the identification effort involved is directly proportional to N.

1.2 SPEECH PRODUCTION MECHANISM

In order to understand various analysis techniques in this field, it is appropriate if the study is preceded by an explanation of the speech production mechanism in human beings. A brief but exhaustive discussion appears in [13] of which the following is an extract.

In general, a statement uttered by a human consists of speech as well as silence. The speech waveform is an acoustic pressure wave produced by voluntary movements of the vocal structures shown in Figure 1. For production of speech, first air is expelled from the lungs into the trachea and then forced between the vocal folds. Speech cosists of voiced and unvoiced sounds. During the production of voiced sounds such as/i/[1]in *eve*, the air from the lungs causes the vocal folds to open and close at a rate dependent upon the air pressure in the trachea and the adjustment of the vocal folds which includes changes in

---

[1] The symbol /./ is used to denote the phoneme, a basic linguistic unit. In this section the phonemes of the General American dialect are used.

length, thickness and tension. The pitch or fundamental frequency of the voice increases with an increase of the tension of vocal folds. The opening between the vocal folds is called the glottis.

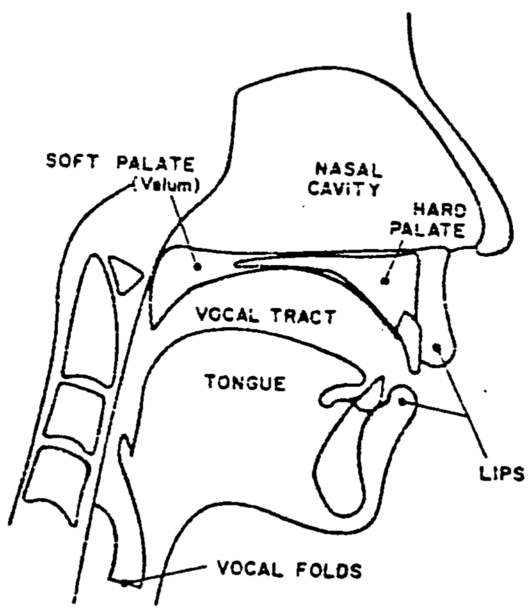


Figure 1: Some of the major anatomical structures involved in speech production.

The vocal tract is a non-uniform acoustic tube which extends from the vocal folds to the lips and varies in shape as a function of time. These variations are caused due to changes in positions of the lips, jaw, tongue and velum. During the production of non-nasal sounds, the velum closes off the vocal tract from the nasal cavity. The nasal cavity is another acoustical tube responsible for the generation of nasal sounds like /n/, /m/ and /ŋ/ as in ru*n*, ru*m* and ru*n*g respectively.

Unvoiced sounds such as /f/ in *f*ish are generated by holding the vocal folds fully open, forcing air through them and then using articulators to create a constriction (such as setting the upper teeth on the lower lip for fish). With both a constriction and vocal fold vibration, voiced fricatives such as /v/ in *v*an are generated and plosive sounds such as /p/ in *p*ot are generated by building up air pressure in mouth and suddenly releasing the air.

The acoustic waveform of the utterance "linear prediction" is shown in Figure 2(A), [13]. Figures 2(B) and 2(C) represent respectively the waveforms from a voiced and an unvoiced portion of this utterance.

It may easily be seen that the waveform in 2(B) representing voiced sound is nearly periodic. The distance T between two major peaks is called the pitch period and is the period of the glottal vibrations. The waveform in 2(C) does not exhibit such a pitch periodic behaviour since it represents the unvoiced sound /ʃ/ (as in predic*tion*) produced by the vocal folds fully open and air stream pushed through a constriction between the tongue and teeth.

Ideally, it is desirable to have the speech production models that are linear and time invariant. But the acoustic waveform has such a complex structure that it does not satisfy either of these properties. Speech is continually time varying and the coupling of glottis and the vocal tract makes it non-linear. However with certain assumptions, it is possible to approximate speech over short time intervals by linear, time invariant models.

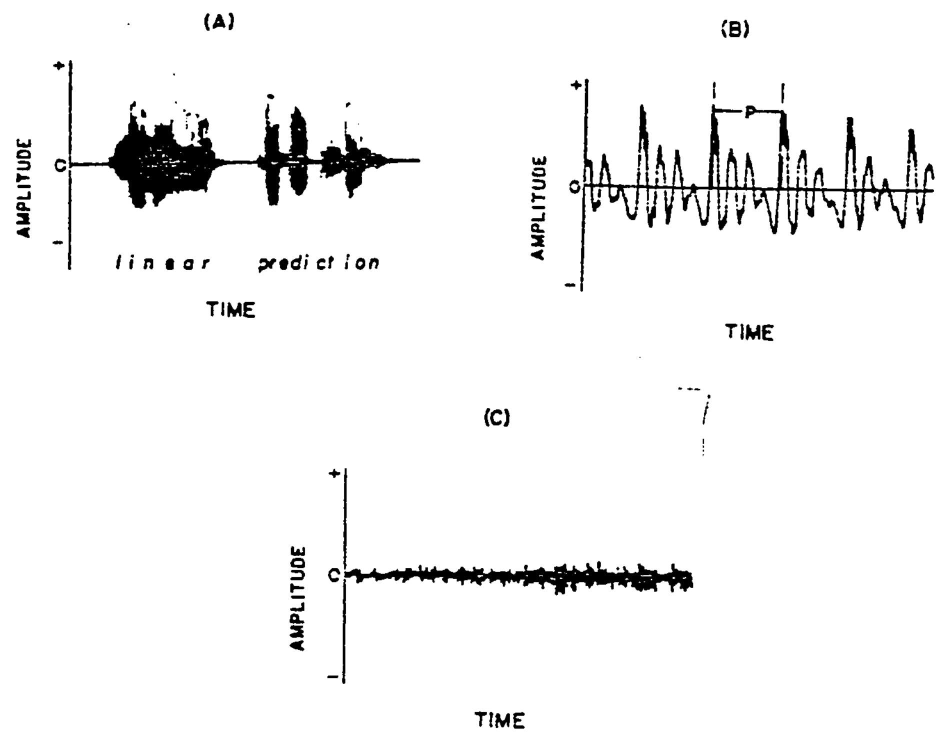


Figure 2: Time domain representations of
(A) the uttrance "linear prediction",
(B) a voiced portion, and
(C) an unvoiced portion.

1.3 HISTORICAL BACKGROUND

For quite a long time, it has been fascinating for man to make machines talk and listen. This interest stems from as far back as ancient Greek and Roman civilizations [14]. The ability of recognizing a person only from his voice is known as speaker recognition.

The early work on the topic of speaker recognition was totally limited to human listening. A part of this research gave rise to vocoders which are machines to synthesize speech. Although the synthesized speech is sufficiently intelligible, it is often deficient with respect to speaker recognizibility [1]. This problem enhanced interest in a search for those parameters in speech that are representative of the speaker rather than the utterance itself. Although with a varying degree of accuracy, machines have been made to recognize speakers, the fundamental question of "how human listeners differentiate between speakers?" still remains unsolved and the afore-mentioned parameters are not fully understood.

With the advent of digital computers, a great impulse has been received by research in the area of speaker recognition and a large number of algorithms have been developed.

One aspect of the use of computer for speaker recognition is very interesting; while in many areas of research machine could not duplicate the accuracy exhibited by human, in this particular area of speaker recognition, machine has surpassed human performance.

1.4 LITERATURE SURVEY

Although automatic speaker recognition is a relatively new subject, extensive research has been carried out during the last two decades due to its promising usefulness specially in the fields of business and criminalogy. As a result a cosiderable amount of literature on this topic is available. As has been pointed out earlier in Section 1.1, the problem of speaker recognition has a number of variations depending upon various factors (such as noisy or noisefree case, test speaker's cooperation etc.) and the available literature can be divided into a corresponding number of categories.

In the beginning, research was carried out on speaker recognition using three dimensional plots of speech energy versus time and frequency. These plots are called spectrograms or voiceprints. Probably the most important piece of literature on speaker recognition was the one that introduced the idea of spectrograms [15]. This importance has been judged from the amount of research stimulated by this paper. It has been reported [12] that the error rates using spectrograms for speaker recognition varied greatly from 0.5% to in excess of 20% [15-19], mostly due to the varying experimental environments and tasks. It was observed that the best results were obtained when it was required only to sort the customer population into a known number of speaker sets where the number of speakers in each set was known and the utterance was fixed. The poorest results were exhibited by those experiments trying speaker identification to the extent that sometimes no identification was obtained [12].

Clearly, this magnitude of error rate was totally unacceptable for the use of speaker recognition in legal courts where it was hoped to replace finger print identification. Moreover, when compared to the results of speaker recognition by human listeners, the error rates for voice print recognition have always been larger [16,18,20]. Human listeners appear to be about 95% successful in recognition of a group of about ten speakers [21-22]. This suggests that some speaker discriminating characteristics utilized by human listeners are not included in voice prints. Efforts have been made to search for the parameters responsible for speaker recognition by human listeners [23], but the psycho-acoustic dimensions of clarity, roughness, magnitude, and animation offer very little hints of physical correlates [12]. Another major drawback in using voice prints is that an expert is needed for their use as is the problem with finger prints.

Ideally, an effective speaker recognition algorithm should measure some aspect of speech that reflects the unique properties of the speaker's vocal system and contains no information about the speech itself.

An excellent survey of various speaker recognition techniques has been carried out by Rosenberg [4]. Speaker recognition is a typical pattern recognition problem which usually consists of two stages:

1. Training stage: where patterns coming from known classes are studied. Efficient parameters are extracted and transformed to a new space to reduce the dimensionality, increase the inter-class distances and decrease the intra-class distances. These reduced transformed parameters are then stored.
2. Recognition stage: where patterns coming from unknown classes are processed in a similar fashion and then the resulting reduced transformed parameters are compared with those of known ones.

Therefore, the important factors in a pattern recognition problem are (i) data collection, (ii) parameter (or feature) extraction, (iii) transformation and dimension reduction and (iv) distance measurement (measurement of dissimilarity).

DATA COLLECTION: There are several factors in which various algorithms differ at the data collection stage. One of these is whether or not there is a prescribed text for the speakers to utter. The groups in which the speaker recognition procedures are divided on the basis of this criterion are respectively known as text dependent and text independent. Most of the speaker recognition procedures are text dependent. The text independent procedures are, however, not completely free from any restrictions on the nature or contents of the utterance text. Usually the restriction imposed on the utterances is that of length [24-27]. Another more general class of procedures necessitates the presence of certain specified speech events in the text which are then extracted and used in the recognition procedure [5,6,28-30]. Such procedures require detection of these events and segmentation of speech based on this detection. The procedures based on visual analysis of voiceprints [15] are text independent in nature with restrictions on the length and content of the utterances.

FEATURE EXTRACTION: The next stage after data collection is the preprocessing of the raw data. At this stage a number of speech features are extracted. Some of the procedures require a filter-bank for this purpose. A filter-bank is a series of band-pass filters covering the useful frequency range of speech. The output of each filter is processed and sampled at a suitable rate to obtain the energy of speech over the filter pass band. Although filter-banks can provide a highly efficient and comprehensive analysis of speech, there is an inherent disadvantage in their use alone. This is because, once the speech is passed through a filter-bank, all subsequent analyses depend upon combining measurements of individual filter outputs. This makes procedures like pitch and formant analysis difficult to achieve through filter-bank processing alone. Hence, in most of the methods that utilize filter banks, other analyses like pitch, formant and overall energy estimations are also included in preprocessing [5,28]. The methods that do not rely on filter-bank techniques make use of pitch, formant, intensity and LPC analysis [6,10,31-32].

In the case of using dynamic features for speaker recognition, the time functions are brought into time registration with reference functions. The results of certain experiments [3] show that there is only a slight difference between the recognition accuracies for statistical features and dynamic features. Since the amount of calculation necessary for recognition using statistical features is only about one-tenth of that for recognition using dynamic features, it is more sensible to use statistical features than dynamic features [3].

A large class of procedures requires segmentation of the input speech. This class contains both text dependent and text independent procedures. In case of text dependent speaker recognition procedures, segmentation of speech is required to precisely align the occurences of similar text events in reference and text utterances. This process makes it possible to compare equivalent events and also compensate for the variations in repetitions of an utterance by a given speaker. This type of segmentation is known as time registration. Usually, time registration is accomplished manually, but some automatic procedures have been investigated [12,31,33-35].

The technique of dynamic time warping for time registration has found widespread use in the area of speaker recognition. In this technique the recognition is based upon a matching process in which the unknown input pattern of coefficients is compared with stored reference patterns. The purpose of the time warping algorithm is to provide a mapping between the time indices of the reference and test utterances such that a time registration between them is obtained [36]. The use of dynamic programming techniques provides an efficient algorithm for nonlinear time alignment as shown by Sakoe and Chiba [37].

Segmentation is also required in the case of text independent recognition procedures which rely upon the extraction of particular acoustic features and events. Here segmentation is used to locate and isolate these events which are found to be strong speaker discriminators. The effectiveness of a large number of these acoustic features have been studied [5,6]. The study of Das and Mohan [28] utilized 405 different features, some of which were filter averages, formant data and pitch frequency. Although the recognition procedures based on segmentation are highly successful and powerful, segmentation itself is an extremely difficult and complicated process. Most of the investigations have used manual segmentation schemes for feature extraction. An automatic scheme has been reported [28] but it can be considered only as a partial success because 10% utterances were discarded due to segmentation failure.

A subclass of procedures using speech segmentation are those which are based on the analysis of nasal consonants [5,6,29-30]. It is hypothesized that the acoustical properties of nasal consonants have strong speaker discrimination potential and that there is little movement of the articulators during phonation. Most of these procedures use the spectral characteristics of individual consonants. One investigation, however, is based on the hypothesis that differences in the spectra of a particular nasal consonant due to coarticulation with the following vowels is not only a strong speaker discriminator but also less subjected to mimicry [30].

In another type of analysis, it is hypothesized that many acoustic features as functions of time are strongly speaker dependent. Some of these features that have been investigated are pitch, formant, intensity and LPC coefficients [10,12,31-32,38]. A comprehensive survey of different speech analysis techniques and various parametric representations of speech appears in [39].

DIMENSION REDUCTION: After preprocessing, analysis is done in order to reduce the dimensionality of the sample space such that the speaker recognition capabilities are preserved or possibly enhanced. Reduction of dimensionality is mostly carried out, experimentally, by discarding parameters having less discriminatory power. This step, however, is not required by those methods which base the recognition task directly on the preprocessed data. Examples of such procedures are [7,40-42] where statistical decision techniques are used directly on spectral energy matrices obtained as filter-bank outputs. The most obvious and simplest analysis technique is the formation of long term averages of the parameters obtained at the outset of preprocessing stage. With filter-bank preprocessing, long term spectral energy averaging can be found in [25-26,41,43]. Other techniques are conversion to cepstral measurements [44] and formant extraction [5,12,31].

Sambur [45] tested the speaker recognition potential of a set of orthogonal parameters, called linear prediction orthogonal parameters. These parameters are formed by a linear transformation of the linear prediction parameters. This set of parameters is essentially independent of all linguistic infomation across an analysed utterance, but highly indicative of the identity of the speaker [45]. The results of these experiments showed an identification accuracy of over 99 percent for high quality speech inputs. Sambur's procedure has a vital advantage in that it does not require any dynamic time warping which is a computationally expensive phenomenon.

A comparison of Sambur's technique and three other techniques has been carried out by Wohlford, Wrench and Landell [46]. Their comparison shows the superiority of Sambur's technique over the other three. This technique has been reported to have 95 percent recognition accuracy as a text-independent recognizer with 10 minutes of reference speech and 30 seconds of test speech. Schaffer and Rabiner [39] have described the procedures based upon the principle of linear prediction as being among the most useful methods of speech analysis. It has been pointed out that these methods are important because of their accuracy and speed of computation.

For some algorithms, statistical feature selection follows the analysis of speech data. Many of the algorithms utilize the F-ratio or analysis of variance for decision taking [32,41]. For the estimation of F-ratio, several statistics including mean feature vector and the covariance matrix of its elements are computed over the set of reference utterances provided by each speaker. Next, a between-talker covariance matrix B, which is a dissimilarity measure of the feature vector means, and a within talker covariance matrix W, which is the average of the individual talker covariance matrices, are computed. Then a set of eigenvectors $a_1, a_2, \ldots$ is used to to transform linearly the original feature vector into a space where the speaker feature vector means are maximally separated relative to the within talkers covariance [4]. To get this set of eigenvectors, the F-ratio given by:

$$F = \alpha^T B \alpha \; / \; \alpha^T W \alpha$$

is maximized over α. As a special case, if the components of the feature vector are assumed to be uncorrelated, by equating the diagonal elements of B and W to zero, the F-ratio provides an ordering of the separability of the between speaker means in terms of the original individual coordinates [5,7,28,48-49].

In another statistical feature selection technique, those features are chosen which provide the lowest error rate from processing a set of reference utterances through the designated speaker recognition system [6,10]. If the original feature vector has a relatively high dimensionality, it is important to investigate as many lower order subsets of this vector as possible. One efficient procedure for this purpose is the "knockout tournament" [6,10].

DISTANCE CALCULATION: Decision techniques are all based on the calculation of a distance which measures the dissimilarity between points in the multi-dimensional feature vector space. Out of a number of distance metrics, the most common and one of the simplest is the Euclidean distance. Let ν_i denote the feature vector corresponding to the ith. reference speaker and ν be the test feature vector. Then the Euclidean distance is given by:

$$d = [(\nu_i - \nu)^T(\nu_i - \nu)]^{\frac{1}{2}}$$

$$= [\sum_j (v_{ij} - v_j)]^{\frac{1}{2}},$$

where v_{ij} and v_j represent the jth. component of ν_i and ν respectively. Other possible metrics are $\sum_j |v_{ij} - v_j|$ and $\nu.\nu_i$ / ($\|\nu\|$ $\|\nu_i\|$) which is the cosine of the angle between the vectors ν and ν_i. Here $|.|$ and $\|.\|$ are used to denote absolute value and vector magnitude respectively. A number of comparative studies of distance measures have been carried out in [25-26,50].

DECISION TECHNIQUES: Once distances have been calculated, the simplest decision rule is that of the "nearest neighbour". This means that distances are calculated from the unknown vector to each vector in the reference set. The speaker whose reference vector corresponds to the minimum distance is chosen. In case of speaker identification, the test speaker is identified as this speaker. For speaker verification, the minimum distance is compared to a threshold value to decide whether to accept or reject. This approach has been adopted by many investigators [25,40,44]. However, modifications to this approach are possible. The nearest neighbour, in one such modification, is based upon calculating the mean feature vector for each reference speaker and then taking decisions using these mean vectors. This decision is evidently more efficient than the more general search over both speaker and utterances. This approach has been used by Bunge [50], Furui et. al. [25], Glenn and Kleiner [29] and Hair and Rekieta [48-49]. In a more advanced technique, the components of the feature vector are weighted before averaging in the above approach. This weighting is done inversely by the calculated variances of these components over each individual speaker's utterances [5,7,10,31,35,38,45,48-50]. The effect of this weighting is to give more influence to those components of the feature vector which are more strongly clustered.

1.5 PROBLEM STATEMENT

All the speaker recognition algorithms that have been discussed in the previous sections are observed to yield reasonably good to excellent performance provided that the speech samples both in the customer set and uttered by the test speaker have been recorded in a noise free environment on high quality equipment. This is a rather unrealistic requirement, because in daily life noise is unavoidable. To illustrate this consider an example. The greatest use a successful speaker recognition algorithm is expected to find is in business transactions. Now it is obvious that real shortcomings for other ways of personal identification arise when the transactions are carried out by telephone. There is a large number of sources of noise associated with the present telephone system. One inherent source is the frequency limitations of speech over telephone. The speech is transmitted only in the lower band of frequencies which gives rise to distortion. Secondly, it is nearly always the case that telephone calls are made in the noisy environment of public call booths situated near streets and roads or at homes or offices where other sources of noise are present. A third cause of noise is due to defective telephone lines where cross-talks are generated.

To illustrate further, take the example of another expected application, namely, that of personal identification by speaker recognition in criminal courts. Assume that the speech samples of criminals are kept on record in a certain country. These samples are collected in recording rooms and are thus noise free. Suppose that a crime is committed and the voice of the criminal has been recorded. Most likely this sample will contain noise generated by various sources. Now the problem is to identify the criminal from his noisy speech while the reference samples are noise free.

With these two examples, it is hoped that the reader would be able to realize the importance of the solution to the speaker recognition problem in a noisy environment. Most of the algorithms that can be found in the literature are observed to yield reasonably good to excellent performance with high quality and relatively noise free speech. Contrary to this, in daily life, noise is unavoidable. Hence it is of interest to study the effect of inclusion of noise. Preliminary studies as reported in literature indicated severe deterioration in the performance of the speaker recognition algorithms in the presence of noise.

In this thesis, the problem of recognizing speaker from noisy speech has been studied. The features (parameters) chosen for this purpose were the LPCs due to their superior performance in speech analysis [39]. To be more specific, Sambur's algorithm [45] that utilizes orthogonal LPC parameters has been chosen because of its great simplicity, as it does not require the computationally expensive procedure of dynamic time warping but still yields high recognition accuracy [46]. It is well known [51] that the conventional estimates of LPCs become biased in presence of noise. In this work various estimation algorithms have been tried to eliminate the bias in LPCs and their effectiveness in noisy environment is studied. For this study, the noise is assumed to be additive and white.

1.6 ORGANIZATION OF THE THESIS

The text of this thesis has been divided into five chapters and a number of appendices. Chapter 1 presents an introduction to some basic concepts in the subject of speaker recognition. A historical background, a literature survey and the importance of the chosen problem are included. In addition, the choice of the methodology is justified.

Chapter 2 deals with the subject of speaker recognition using LPC parameters in detail, which is divided into a number of sections. The first two sections are about the linear prediction modelling of speech and estimation of the parameters. The next two sections are about the application of the linear prediction coefficients (LPCS) and orthogonal LPCs (OLPCs) to speaker recognition. The last section has been devoted to a summary of the chapter.

The third chapter discusses the problems of speaker recognition in a noisy environment. The first section compares statistical properties of LPCs in noise free and noisy cases. The second and third sections suggest various procedures to reduce the degrading effects of noise. In Section 3.4, speaker recognition using these procedures is discussed followed by the last section that summarizes these ideas.

Chapter 4 is based on the discussion of the experimental environments and the results obtained. The first section discusses the creation of the database used in this study. Then the results for the cases of clean and noisy speech are presented in a number of sections and subsections. The last section presents a summary of these results.

The last chapter consists of a discussion of the results presented in Chapter 4. Also included are the conclusion and the recommendations for future research.

Chapter II

SPEAKER RECOGNITION USING ORTHOGONAL LPC PARAMETERS

2.1 INTRODUCTION

In this chapter the estimation of LPC parameters and their use in speaker recognition procedures will be discussed. Before going into the details of LPC coefficients, a brief introduction to various speech parameters is presented. These parameters have been found to be useful in various areas of speech research.

2.1.1 SOME SPEECH PARAMETERS

1. *Intensity:-* One of the most obvious and easiest characteristics of any signal is its intensity, which is a function of time for non-stationary signals like speech. An appropriate expression for intensity of a signal s(t) over a time interval (t-T/2,t+T/2) is

$$E(t) = \int_{t-\frac{T}{2}}^{t+\frac{T}{2}} s^2(\tau)\, d\tau/T.$$

The choice of the averaging interval length T is somewhat arbitrary; 10 to 30ms being suitable in most cases [1]. The variations in intensity of sound are caused both by the subglottal pressure as well as changes in the vocal tract shape as a function of time. Intensity is an important source of speaker dependent information in speech [31,52].

2. *Pitch:-* Pitch is defined as the fudamental frequency of the vocal cord vibrations. Accurate measurement of pitch is not simple and has been extensively investigated. In the time domain, pitch can be determined by measurement of the period of the speech waveform and in the frequency domain by computing the frequency spacing of the spectral peaks [53-57]. The temporal variation of pitch represents an important characteristic of pitch and has been found to be useful for automatic speaker recognition [32,53].
3. *Short-Time Spectrum:-* Short-time spectrum provides a three dimensional representation of speech, the dimensions being energy, frequency and time [58]. The short-time spectrum is a complete representation of the acoustical characteristics of speech, although this representation is not very compact [20]. Both the exact short-time spectrum as well as its approximation by filter bank outputs have been found to be effective in speaker recognition [24,28,40-41].
4. *Linear Prediction Coefficients:-* A detailed discussion of LPCs will be found in the next section. It is an important representation of spectral properties of speech in the time domain [39,59-62]. In this method each sample of a signal is assumed to be predictible from weighted linear combinations of past inputs [61]. This is the the reason for its name. The

weights associated with the minimum value of the mean squared prediction error are called linear prediction coefficients or LPCs. If the speech is band limited to 5 KHz., typically, 12 predictor coefficients are adequate. LPCs are functions of time with such small variations that it is sufficient to calculate them once in every 15 to 30 ms.

5. *Formant Frequencies and Band Width:-* Formant frequencies are the resonance frequencies of the vocal tract. Although they are speaker dependent, their measurement is so difficult that they have not gained popularity. Several methods may be found in the literature [63-66] that give partial solutions to the problem of determining formants. Nevertheless, accurate determination of formant frequencies poses very difficult problems.

6. *Spectral Correlations:-* A significant degree of correlation exists between the short time spectrum at different frequencies [1]. These correlations are speaker dependent and are found to vary from speaker to speaker. Stable evaluation of these correlations is, however, a difficult task and requires averaging over at least 30 seconds of speech.

7. *Timing and Speaking Rate:-* Relative timing and duration of utterance of speech events in the same statement are speaker dependent. A way of determining these differences has been suggested by determining the non-linear deformation of the time axis of one utterance relative to another [12,67].

2.1.2 LPC PARAMETERS

Linear predictive coding (LPC) is a highly successful modelling technique that is based upon the assumption that the combined effect of the various factors of speech production such as the glottal source, the vocal tract etc. can be represented as an all pole (or autoregressive) digital filter. This approach carries out the prediction of the kth. sample of speech as a linear combination of the previous p samples, where p is known as the order of the linear prediction. Thus if x_k represent the kth. sample, then

$$x_k = \sum_{i=1}^{p} a_i \, x_{k-i} + \varepsilon_k, \qquad (2.1.1)$$

where ε_k is the error in prediction. Equation (2.1.1) is a relation for LPC. The weighting coefficients $\{a_i\}$ are characteristics of the filter and are known as the LPC parameters LPC coefficients or simply LPCs.

The method of least squares computes these coefficients so that the mean squared error between the predicted speech and the actual speech is minimized. Since speech has a highly dynamic behaviour, the LPC parameters can only be satisfactorily estimated if a sufficiently small part of speech is considered that is approximately stationary[1]. This is possible because although speech has a time varying behaviour, these variations are slow and continous. Experiments have shown that a part of speech corresponding to a duration of 15 to 30 milli-seconds is appropriate for the purpose of LPC parameter estimation.

---

[1] A signal is said to be stationary if i) it has a constant mean and ii) its autocorrelation function is a function of only the lag.

2.2 ESTIMATION OF LPCS

The predictor coefficients $\{a_i\}$ can be found by solving a set of p linear equations simultaneously. Out of a number of procedures for the estimation of LPC parameters, a computationally efficient time domain algorithm i.e. the auto-correlation approach will now be discussed.

AUTOCORRELATION APPROACH FOR LPC PARAMETER ESTIMATION:

Let the speech be divided into frames, each of length L samples so that the stationarity condition as described earlier (Section 2.1.2) is satisfied. Define an infinite sequence $\{s_k\}$ by:

$$\begin{aligned} s_k &= x_k \qquad \text{if } 1 \le k \le L, \text{ and} \\ &= 0 \qquad \text{otherwise.} \end{aligned} \tag{2.2.1}$$

Equation (2.1.1), then, becomes:

$$s_k = \sum_{i=1}^{p} a_i \, s_{k-i} + \varepsilon_k \, .$$

Assuming that $a_0 = -1$, this equation is simplified to

$$\varepsilon_k = \sum_{i=0}^{p} a_i \, s_{k-i} \tag{2.2.2}$$

Let TSE denote the total squared error as k ranges from $-\infty$ to $+\infty$, then:

$$TSE = \sum_{k=-\infty}^{\infty} \varepsilon_k^{2}$$

$$= \sum_{k=-\infty}^{\infty} \{ \sum_{i=0}^{p} a_i \, s_{k-i} \}^2$$

$$= \sum_{k=-\infty}^{\infty} \{ \sum_{i=0}^{p} \sum_{j=0}^{p} a_i \, a_j \, s_{k-i} \, s_{k-j} \}$$

$$= \sum_{i=0}^{p} \sum_{j=0}^{p} a_i \, a_j \sum_{k=-\infty}^{\infty} s_{k-i} \, s_{k-j} \, .$$

But the autocorrelation function $r_{ss}(j)$ of the sequence $\{s_i\}$ is defined by

$$r_{ss}(j) = \sum_{i=-\infty}^{\infty} s_i \, s_{i-j} \qquad (2.2.3)$$

Hence,

$$TSE = \sum_{i=0}^{p} \sum_{j=0}^{p} a_i \, a_j \, r_{ss}(i-j) \qquad (2.2.4)$$

The values of $\{a_i\}$ can, then, be found by minimizing the total squared error over a_k, k=1,2,..,p. This requires:

$$\partial \, TSE \, / \, \partial \, a_k = 0, \qquad k=1,2,..,p.$$

From equation (2.2.4), this gives:

$$\sum_{i=0}^{p} a_i \, r_{ss}(i-k) = 0, \qquad k=1,2,..p, \quad (2.2.5)$$

since, from (2.2.3), $r_{ss}(j)$ is a symmetric function.

Now, setting $a_0 = -1$ in (2.2.5) gives:

$$\sum_{i=1}^{p} a_i\, r_{ss}(i-k) = r_{ss}(k), \qquad k=1,2,..,p. \quad (2.2.6)$$

(2.2.6) is a set of p linear equations in p unknowns $\{a_i\}$, the solution of which yields the desired LPC parameters.

Equations (2.2.6) can be solved by a large number of different procedures, such as the Gauss elimination method and the Crout reduction method. These methods are general in nature and require more than $p^3/3$ multiplications and divisions and p^2 storage locations. The special structure of equations (2.2.6), however, allows for more efficient algorithms for their solution. It can be easily noted that the coefficient matrix of the system of equations (2.2.6) is not only symmetric but also positive semi-definite, as this is a covariance matrix. Covariance matrices are in general positive semi-definite, although in practice they are usually positive definite [62]. Moreover, the right hand sides of these equations are themselves included in the coefficient matrix. This special structure gave rise to an excellent algorithm by Durbin [47]. This algorithm is recursive in nature and is presented in Appendix A.

2.3 *SPEAKER RECOGNITION USING LPC*

It has been shown [68] that linear predictive coding is a great aid in providing a fast and reliable procedure for measuring the speaker dependent features in speech waveform. However, in that study the LPC parameters were not directly utilized for speaker recognition. Rather, the parameters were used to determine steady state formant and bandwidth data during an ensemble of speech events. The extraction of steady state formant and bandwidth data was based upon a time consuming polynomial root finding process that was not error free.

These problems in the indirect usage of the LPC parameters led to the question of whether the conversion from LPC coefficients to formant and bandwidth data was really necessary to obtain speaker characterizing features. An experiment was performed [31] to compare the effectiveness of these two approaches in speaker recognition. This experiment showed that the absolute performance of the approach using LPC parameters directly was slightly better than the previous indirect approach. Moreover, the computation time was reduced by half when the direct approach was used.

In the above speech dependent procedure, the speech signals were segmented into frames. The LPCs for the frames were estimated yielding a time contour for each LPC coefficient of an utterance. Reference contours were made by time aligning and averaging a set of utterances from the same speaker speaking the same phrase. The test utterance was then time aligned and compared with the reference. A few of the coefficient contours (which were selected through extensive experimentation) were found to have good speaker discrimination power.

Nevertheless, the algorithm suffers from computationally expensive time alignment requirements and arbitrary selection of parameter contours.

2.4 *SPEAKER RECOGNITION USING OLPC*

INTRODUCTION: Orthogonal linear predictive coding (OLPC) is a technique which yields a set of orthogonal parameters by an appropriate eigenvector analysis of LPC coefficients. These parameters have been shown to be highly indicative of the talker's identity but invariant with respect to the linguistic contents of the speech. A great advantage in the use of the OLPC parameters is that the complicated, computationally expensive and error prone operations of time normalization and time warping are not required. Even then, the recognition accuracy of this procedure has been shown to be better than others [45].

In other methods, where the selected recognition feature is indicative of both the speaker and the text of speech, full discrimination potential can only be achieved if the effect of the speech content is completely eliminated. This is where these methods require time normalization and segmentation of the utterances. The LPC procedure of speaker recognition is one of these methods.

The technique of orthogonal linear prediction was introduced to exploit the experimental observation that the LPC parameters were considerably redundant [45]. It can be easily implied from this observation that a conventional eigenvector analysis can be used to reduce the dimensionality of the LPC space. With this analysis a set of statistically uncorrelated (orthogonal) parameters are produced that are formed by a linear combination of the LPC parameters. These parameters are, therefore, decoupled in the sense that their covariance matrix is diagonal. This is why they are known as orthogonal LPCs.

The OLPC parameters were first used in the domain of speech synthesis. An experimental study [69] showed that these parameters could be used to achieve high quality speech synthesis. Their use for speaker recognition, however, initiated from the fact that a large subset of these parameters demonstrated an invariance across the analyzed utterance. These invariant parameters were found to be completely specified by their measured mean values across the utterance. It was therefore clear that these parameters characterized some attribute of speech that was unchanged over the utterance. One such attribute was, of course, the speaker himself. Hence, it was hypothesized that the parameters under discussion could be identifying the speaker of the utterance. A second experiment strengthened this hypothesis. This experiment showed that when the same eigenvector analysis was applied to two utterances of the same statement spoken by two different speakers, the resulting mean values of the orthogonal parameters under discussion were different. It was, therefore, clear that a set of orthogonal parameters could be found that was free from any linguistic information but highly indicative of the speaker's identity. The set of OLPC parameters was, then, actually examined for their speaker recognition potential [45]. This experiment was performed on a 21 speaker population. An accuracy of 96.8% was observed when 12 OLPC parameters were considered for speaker identification. When only 4 parameters were used, the accuracy was still 95.8%. The results for the verification case showed an accuracy of 95.2% for 12 parameters and 90.4% for 5 parameters.

DERIVATION: Suppose the number of parameters to be found is p. Let L be the number of utterances used for constructing the reference set of parameters for a speaker in the customer set and let the lth. utterance be divided into J_l frames of appropriate length so as to satisfy the stationarity conditions. Then the covariance matrix R_l of the LPC parameters for the lth utterance of this speaker is given by:

$$R_l = [\ r_{ik}\ ]_{p \times p} \qquad (2.4.1)$$

where:

$$r_{ik} = \{ \sum_{j=1}^{J_l} (a_{ijl} - \bar{a}_{il})(a_{kjl} - \bar{a}_{kl}) \} / (J_l - 1) \qquad i,k=1,2,..,p$$

Here a_{ijl} (i=1,2,..,p) represents the ith. LPC coefficient in the jth. frame of the given utterance of the mth. speaker. $\bar{a}_{il}$ is the mean value of the ith LPC parameter over all the frames of the utterance, i.e.,

$$\bar{a}_{il} = (\sum_{j=1}^{J_l} a_{ijl}) / J_l \qquad (2.4.2)$$

When these covariance matrices for all the L utterances of the speaker have been found, the unique covariance matrix $\bar{R}$ can then be evaluated as the weighted average of the matrices R_l. Hence,

$$\overline{R} = \left(\sum_{l=1}^{L} J_l R_l \right) / \left(\sum_{l=1}^{L} J_l \right) \tag{2.4.3}$$

Let $\{\lambda_i\}$ and $\{\boldsymbol{b}_i\}$, i=1,2,..p, respectively denote the eigenvalues and the corresponding mutually orthogonal eigenvectors of $\overline{R}$. $\{\lambda_i\}$ can be found by solving the pth. degree algebraic equation:

$$\det(\lambda I-\overline{R}) = 0,$$

for λ, where det(.) denotes the determinant of a matrix. The eigenvectors of $\overline{R}$ are then obtained as the solutions of

$$\lambda_i \boldsymbol{b}_i = \overline{R} \boldsymbol{b}_i \qquad i=1,2,..p.$$

Let ϕ_{ijl} denote the ith. OLPC parameter in the jth. frame of the speaker's lth. utterance, then ϕ_{ijl} is given by:

$$\phi_{ijl} = \sum_{k=1}^{p} b_{ik} a_{kjl} \tag{2.4.4}$$

where b_{ik} denotes the kth. component of the ith. eigenvector. The average value of the ith. orthogonal parameter is then:

$$\overline{\phi}_i = \sum_{l=1}^{L} \sum_{j=1}^{J_l} \phi_{ijl} \Big/ \sum_{l=1}^{L} J_l \tag{2.4.5}$$

VARIANCE OF OLPCs: Let the eigenvalues $\{\lambda_i : i=1,2,..,p\}$ of $\overline{R}$ be so arranged that $\lambda_1 > \lambda_2 > .. > \lambda_p$. Let b_{ij} denote the jth. component of the ith. eigenvector b_i. Define the eigenvector assembly matrix B of $\bar{R}$ by $B^T=[b_{ij}]_{p\times p}$. Also assume that:

$$\Phi_{jl} = [\phi_{1jl},\phi_{2jl},..,\phi_{pjl}]^T.$$

Then from equation (2.4.4) it follows that:

$$\Phi_{jl} = B^T a_{jl} \qquad (2.4.6)$$

The covariance matrix of Φ_{jl} is given by:

$$\begin{aligned} \mathrm{cov}(\Phi_{jl}) &= E\{(\Phi_{jl}-\bar{\Phi})(\Phi_{jl}-\bar{\Phi})^T\} \\ &= B^T E\{(a_{jl}-\bar{a})(a_{jl}-\bar{a})^T\}B && \text{using (2.4.6)} \\ &= B^T\bar{R}\ B && \text{from the definition of R.} \end{aligned}$$

where $\bar{\Phi}$ and $\bar{a}$ have been used to denote the averages of the vectors Φ_{jl} and a_{jl} over all the reference utterances of the mth. speaker. But $\bar{R}$ is a real, symmetric matrix of which B is the eigenvector assembly matrix. Therefore:

$$B^T\bar{R}\ B = \mathrm{diag}(\lambda_1,\lambda_2,..,\lambda_p).$$

This implies that:

$$\mathrm{cov}(\Phi_{jl}) = \mathrm{diag}(\lambda_1,\lambda_2,..,\lambda_p) \qquad (2.4.7)$$

Since $\lambda_i>\lambda_{i+1}$, it may be concluded from equation (2.4.7) that the lagging OLPC parameters are less variant across the speech of a given speaker. And hence they may be expected to bear more speaker discriminating power.

RECOGNITION PROCEDURE: Next is presented the recognition procedure. Let d_m denote the dissimilarity measure between the mth. speaker and an unknown speaker whose utterance is given for test. Let $\bar{Z}_{im}$ denote the mean value of the ith. orthogonal parameter calculated using the LPC parameters of the test utterance via the orthogonal eigenvectors of the mth. speaker. Then a suitable expression for d_m is:

$$d_m = \sum_{i=q}^{p} \{ (\bar{\phi}_{im} - \bar{Z}_{im}) / \sqrt{\lambda_{im}} \}^2 \, \bar{J}_m \qquad (2.4.8)$$

where $\bar{J}_m$ is the average number of frames in the utterances of the mth. speaker, i.e.,

$$\bar{J}_m = \sum_{l=1}^{L} J_{lm} / L,$$

and q-1 is the number of leading orthogonal parameters not to be included in the distance calculation. λ_{im} are the reference eigenvalues of the mth. speaker and $\bar{\phi}_{im}$ denotes the ith. average OLPC parameter in his reference template. This distance measure gives the weighted distance between mean values of the orthogonal parameters calculated from the reference utterances, and the mean values of the orthogonal parameters calculated across the test speaker. It may be noted that the weights used in d_m are the inverse variances of the OLPC parameters so as to give more weight to parameters that are relatively invariant over the reference utterances of the mth. speaker.

Once the value of d_m has been found for each speaker in the customer set, the unknown speaker is identified as the one corresponding to the minimum distance.

2.5 SUMMARY

This chapter begins with an introduction to a number of speech parameters with special emphasis on the LPC coefficients. Next the autocorrelation approach for LPC parameter estimation has been presented. A brief discussion on speaker recognition using LPC parameters is also included. The OLPC approach has been discussed in detail.

Chapter III

SPEAKER RECOGNITION IN NOISY ENVIRONMENT

3.1 STATISTICAL PROPERTIES OF LPC

In Section 2.2, the least squares (LS) method of parameter estimation has been described for the estimation of the LPC parameters. The LS method is successful provided the system is noise-free. If the LS method is applied to a system incorporating noise, the resulting parameter estimates become biased and inconsistent. In such a situation there are other procedures like the generalized least squares (GLS), extended least squares (ELS) and instrumental variables (IV) that can preferably be used to estimate system parameters. These methods will be described in the later sections of this chapter. This section is devoted to illustrate the presence of bias in the LS estimates of noisy system parameters.

3.1.1 BIAS ANALYSIS

NOISEFREE CASE: The LPC model of speech, as described in the last chapter, is given by:

$$s_k = \sum_{i=1}^{p} a_i\, s_{k-i} + \varepsilon_k \quad , \qquad k=p+1,p+2,..,L. \quad (3.1.1)$$

In these equations, s_k represents the clean speech. The term ε_k accounts for the prediction error. L is the frame length.

The bias analysis can be easily performed if (3.1.1) is written in its vector form. For this purpose, define matrix S and vectors α, s and ε by:

$$\alpha = [a_1, a_2, \ldots, a_p]^T,$$

$$s = [s_{p+1}, s_{p+2}, \ldots, s_L]^T,$$

$$\varepsilon = [\varepsilon_{p+1}, \varepsilon_{p+2}, \ldots, \varepsilon_L]^T \text{ and}$$

$$S = [\; s_{p+i-j} \;]_{(L-p)\times p} \;,$$

where $[.]^T$ denotes the transpose of the matrix [.]. With these notations (3.1.1) can be written as:

$$s = S\alpha + \varepsilon. \qquad (3.1.2)$$

The LS criterion is, then, based upon minimizing the total squared error, which is given by:

$$\mathrm{TSE} = \sum_{i=p+1}^{L} \varepsilon_i^2$$

$$= \varepsilon^T \varepsilon$$

$$= (s - S\alpha)^T (s - S\alpha).$$

Let $\hat{\alpha}_{LS}$ denote the LS estimate of α. Then $\hat{\alpha}_{LS}$ is given by [70]:

$$\hat{\alpha}_{LS} = [S^T S]^{-1} S^T s.$$

Substituting the value of s from equation (3.1.2) this becomes:

$$\hat{a}_{LS} = [S^TS]^{-1}S^T[Sa + \varepsilon]$$

$$= a + [S^TS]^{-1}S^T\varepsilon. \tag{3.1.3}$$

Now assuming the speech sample and prediction error as uncorrelated and taking expectation of both sides one gets:

$$E\{\hat{a}_{LS}\} = a + E\{(S^TS)^{-1}S^T\varepsilon\}$$

$$= a,$$

which shows that in the absence of measurement noise the LPC estimates are unbiased.

BIAS ANALYSIS OF NOISY SPEECH: Now consider estimation of LPCs in the presence of noise. Define y_k and v_k as the noisy speech sample and a Gaussian distributed, zero mean, stationary noise respectively such that:

$$y_k = s_k + v_k. \tag{3.1.4}$$

Substituting from (3.1.4) into (3.1.1) one gets:

$$y_k = \sum_{i=1}^{p} a_i y_{k-i} + e_k \tag{3.1.5}$$

where $\quad e_k = \varepsilon_k - \sum_{i=0}^{p} a_i v_{k-i}$ with $a_0=-1$.

The residual e_k now accounts for both the prediction error as well as the additive noise. In addition e_k and y_k are mutually correlated.

Defining:

$$\boldsymbol{y} = [y_{p+1}, y_{p+2}, \ldots, y_L]^T,$$

$$\boldsymbol{e} = [e_{p+1}, e_{p+2}, \ldots, e_L]^T, \text{ and} \tag{3.1.6}$$

$$Y = [\ y_{p+i-j}\ ]_{(L-p)\times p}\ ,$$

the least squares estimate of α that minimizes the sum residual squares is given by:

$$\hat{\alpha}_{LS} = [Y^T Y]^{-1}\ Y^T \boldsymbol{y}. \tag{3.1.7}$$

Using an analogous approach as equation (3.1.3), it can be shown that:

$$\hat{\alpha}_{LS} = \alpha + [Y^T Y]^{-1} Y^T \boldsymbol{e}.$$

Taking expectation of both sides one gets:

$$E\{\hat{\alpha}_{LS}\} = \alpha + E\{(Y^T Y)^{-1} Y^T \boldsymbol{e}\}$$

$$= \alpha + \text{bias}. \tag{3.1.8}$$

Because of the nonzero correlation between y_k and e_k, the bias is nonzero as well.

3.1.2 CONSISTENCY

Consider the correlation version of the LS estimate given by equation (2.2.6). In the limiting case as L approaches infinity the computed correlation functions will approach their true values and therefore the estimated parameters obtained from solution of the correlation equations will converge to unique values. This property together with the unbiasedness property establishes the consistency of autocorrelation method for noise free speech i.e. it implies that as L approaches infinity the estimates converge to the true values.

Now consider the noisy case. The correlation version of the LS estimate given by (3.1.7) will lead to the solution of the following equations:

$$\sum_{i=1}^{P} a_i r_{yy}(i-k) = r_{yy}(k). \tag{3.1.9}$$

In the limiting case as L approaches infinity, the above autocorrelation functions will be related to the autocorrelation functions of the speech as:

$$r_{yy}(j) = r_{ss}(j) + r_{vv}(j). \tag{3.1.10}$$

The property that the noise is uncorrelated to the speech is used in obtaining (3.1.10). It is evident, therefore, that as L approaches infinity the solution of (3.1.9) will also converge to unique values. However the algorithm is biased. Therefore the solution in the limit will converge to unique values but not to the true parameter values. This implies that the estimator is inconsistent in the presence of noise. This also shows that as L approaches infinity the solution of equations

(2.2.6) and (3.1.9) will yield different parameter values.

3.2 CONSISTENT ESTIMATION OF PARAMETERS

In the previous section it was shown that the LS method of parameter estimation yields biased and inconsistent parameters when the observable output contains noise. In this section various methods to handle this problem will be discussed.

In the field of parameter estimation numerous extensions of LS method have been proposed to overcome the bias. They are the Generalized least squares (GLS), Extended least squares (ELS), Instrumental variable (IV) method, Autocorrelation subtraction (AS) method and Shifted Yule-Walker (SYW) equations. In addition to these procedures we will also examine the speech enhancement algorithms in which an estimate of the clean speech will be obtained through some filtering algorithm and then a least squares algorithm will be applied to obtain the LPC parameters from the estimated enhanced speech.

3.2.1 GENERALIZED LEAST SQUARES METHOD (GLS)

The autocorrelated residual e_k is responsible for the biased estimates of the LPC parameters. To overcome the problem of correlated residuals, the GLS algorithm models the residual by another autoregressive (AR) series and then it estimates the speech parameters and the residual parameters together [51]. Let equation (3.1.5) be rewritten as:

$$A(q^{-1})y_k = e_k \, , \tag{3.2.1}$$

where:

$$A(q^{-1}) = 1 - a_1q^{-1} - a_2q^{-2} - \dots - a_pq^{-p},$$

and q^{-1} is the backward shift operator with the property that:

$$q^{-1}y_k = y_{k-1}.$$

It is assumed that the residual e_k has a rational spectrum such that it may be modelled by the following AR-series:

$$C(q^{-1})e_k = \zeta_k \, , \tag{3.2.2}$$

where:

$$C(q^{-1}) = 1 + c_1q^{-1} + c_2q^{-2} + \dots + c_mq^{-m}.$$

In this definition, c_i are constant coefficients, m is the order of the model and ζ_k is an identically and independently distributed zero mean random sequence. In general, m is not known but is chosen sufficiently high so that (3.2.2) adequately describes the residuals. The algorithm described here estimates the residual parameters c_i together with the process parameters a_i.

Premultiplying both sides of equation (3.2.1) by $C(q^{-1})$ and using (3.2.2) one obtains:

$$A(q^{-1})f_k = \zeta_k \, , \tag{3.2.3}$$

where:

$$f_k = C(q^{-1})y_k.$$

f_k is known as the 'filtered signal' and $C(q^{-1})$ is known as the 'whitening filter'. Note that the residual of (3.2.3) is a white sequence and therefore the conventional correlation algorithm can be applied to estimate the unknown parameters. The algorithm is summarized below:

1. Obtain biased LPC parameters from:

$$\sum_{i=1}^{p} a_i r_{yy}(j-i) = r_{yy}(j) \qquad j=0,1,..,p-1 \quad (3.2.4)$$

2. Estimate $r_{ee}(j)$ using the following equations:

$$r_{ye}(j) = A(q^{-1})\ r_{yy}(j) \qquad (3.2.5)$$

$$r_{ey}(j) = r_{ye}(-j) \qquad (3.2.6)$$

$$r_{ee}(j) = A(q^{-1})\ r_{ey}(j).$$

3. Estimate residual parameters by solving:

$$\sum_{i=1}^{m} c_i r_{ee}(j-i) = r_{ee}(j) \qquad j=0,1,..,m-1 \quad (3.2.7)$$

4. Estimate autocorrelation function $r_{ff}(j)$ using:

$$d_i = \sum_{l=0}^{m-i} c_l c_{l+i} \quad \text{with } c_0=1.$$

$$D(q^{-1}) = d_0+d_1(q+q^{-1})+..+d_m(q^m+q^{-m})$$

$$r_{ff}(j) = D(q^{-1})\ r_{yy}(j) \tag{3.2.8}$$

5. Solve the following equations to obtain speech parameters:

$$\sum_{i=1}^{p} a_i r_{ff}(j-i) = r_{ff}(j) \tag{3.2.9}$$

6. Go to step 2 until convergence.

As the algorithm minimizes the sum squared white residual ζ_k (see equation (3.2.3)), it can be shown that the estimator is consistent [70].

The GLS algorithm is a relaxation procedure to minimize a highly nonlinear criterion. Its convergence to the true parameter values is not guaranteed. For its success, the initial solution should be sufficiently close to the global minimum. It is observed that in practice it does converge to the true parameter values when the signal to noise ratio is high. But at low signal to noise ratio it may converge to a local minimum thus producing misleading estimates [70].

3.2.2 EXTENDED LEAST SQUARES METHOD (ELS)

The ELS algorithm is another extension of the least squares estimator to obtain consistent estimates [71]. This algorithm also assumes a noise model of the residuals, but unlike GLS algorithm it does not require filtering. From equation (3.2.1):

$$A(q^{-1})y_k = e_k. \tag{3.2.10}$$

Let the residuals be described by equation (3.2.2), which may be rewritten as:

$$e_k + C^*(q^{-1})e_k = \zeta_k, \tag{3.2.11}$$

where:

$$C^*(q^{-1}) = c_1q^{-1} + c_2q^{-2} + \ldots + c_mq^{-m}.$$

Substituting equation (3.2.10) in (3.2.11), one may obtain:

$$A(q^{-1})y_k + C^*(q^{-1})e_k = \zeta_k. \tag{3.2.12}$$

Multiplying both sides of equations (3.2.12) by y_{k-j} and taking expectations one obtains [72]:

$$r_{yy}(j) + \sum_{i=1}^{m} c_i r_{ye}(j-i) = \sum_{i=1}^{p} a_i r_{yy}(j-i) \tag{3.2.13}$$

Once the estimates of c_i and $r_{ye}(j-i)$ are available, the above equations (for $j=0,1,..,p$) may be used to estimate the α parameters. In the beginning, initial values for α are obtained by ordinary least squares estimation as the solution of equations (3.2.4). Then the estimates of c_i are obtained through the use of equations (3.2.5), (3.2.6) and (3.2.7).

The estimated values of c_i and $r_{ye}(j-i)$ are substituted in equation (3.2.13) and the values of a_i are obtained by solving (3.2.13).

The algorithm then iterates to improve the estimates of c_i and a_i by use of equations (3.2.5), (3.2.6), (3.2.7) and (3.2.13) respectively. This algorithm is also a relaxation procedure to minimize the sum of squared white residuals and therefore, if converged to the global minimum, will produce consistent estimates. However, there are counter examples where (usually at high noise levels) it may converge to a local minimum yielding misleading estimates [73].

3.2.3 INSTRUMENTAL VARIABLE METHOD (IV)

The IV method is based upon the use of an auxiliary time signal called the instrumental variables. The instrumental variables are defined as a set of variables which are highly correlated with the speech but totally uncorrelated with additive noise terms [74]. Assuming that such an auxiliary stationary time signal z_k is available, an estimation algorithm may be derived as follows. Multiplying both sides of equation (3.2.10) by z_{k-j} and taking expectation of both sides one may obtain [74]:

$$r_{zy}(j) = \sum_{i=1}^{p} a_i r_{zy}(j-i). \qquad (3.2.14)$$

In this equation z_k is the auxiliary time signal and it is assumed that z_k is uncorrelated with v_i and hence with the residual e_k.

Equations (3.2.14), for $j=0,1,..,p-1$, may be used to estimate the unknown parameters a_i. Defining:

$$Z = [z_{p+i-j}]_{(L-p)\times p}\ ,$$

the solution of (3.2.14) may also be written as (see the derivation following equation (3.1.2)):

$$\hat{a}_{IV} = [Z^T Y]^{-1} Z^T \boldsymbol{y}, \qquad (3.2.15)$$

where $\hat{a}_{IV}$ represents the solution of equation (3.2.14), with j=0,1,..,p-1. Y and $\boldsymbol{y}$ are defined in equations (3.1.6).

Using an analogous approach as for equation (3.1.3), equation (3.2.15) may be written as:

$$\hat{a}_{IV} = \boldsymbol{a} + [Z^T Y]^{-1} Z^T \boldsymbol{e}, \qquad (3.2.16)$$

As L approaches infinity (3.2.16) may be written as:

$$\lim_{L\to\infty} \hat{a}_{IV} = \boldsymbol{a} + \{ \lim_{L\to\infty} [Z^T Y]^{-1} \}\{ \lim_{L\to\infty} Z^T \boldsymbol{e}\}. \qquad (3.2.17)$$

The assumption that z_k is highly correlated with the speech and that it is totally uncorrelated with e_k leads to the following relationships:

$$\lim_{L\to\infty} [Z^T Y] = \lim_{L\to\infty} [Z^T S] > 0, \qquad (3.2.18)$$

$$\text{and } \lim_{L\to\infty} [Z^T \boldsymbol{e}] = 0,$$

where S has been defined following equation (3.1.1). Now combining equations (3.2.17) and (3.2.18) one may obtain:

$$\lim_{L\to\infty} \hat{\alpha}_{IV} = \alpha,$$

showing that the estimator is consistent.

Naturally, the question arises as to how one may obtain the above-mentioned auxiliary time signal z_k. Fortunately, the answer is not difficult. It has been mentioned in Section 1.2 that voiced speech is quasi-periodic. Therefore in a voiced frame if one takes the speech samples a pitch period away (delayed or advanced), a time signal is obtained which is highly correlated with the clean speech. In addition as the noise bandwidth is much wider than the speech bandwidth, the delayed (or advanced) signal $y_{k\pm T}$ will be nearly uncorrelated with the measurement noise e_k. Therefore one may write:

$$z_k = y_{k-T},$$

where T is the pitch period. With this definition of the instrumental variables, equation (3.2.14) becomes:

$$r_{yy}(T+j) = \sum_{i=1}^{p} a_i r_{yy}(T+j-i). \tag{3.2.19}$$

Solving the above equations (for j=0,1,..,p-1) for a_i gives the required instrumental variable estimates of α. The solution procedure does not require an explicit solution of the p equations. Instead, the Toeplitz structure of the coefficient matrix may advantageously be applied. A computationally simple algorithm for this purpose is given in Appendix B.

The algorithm, however, requires estimation of the pitch period. Various techniques such as maximum likelihood [75], average magnitude difference [76], parallel processing technique [56] and real time hardware pitch detection [77] may be used for this purpose. Other techniques may be found in [54-55], and a comparative study of various pitch detection algorithms appears in [78]. Of these, the maximum likelihood approach has been adopted in this work (see Chapter 4 and Appendix D). The added computation for pitch extraction, however, makes the algorithm somewhat unattractive.

3.2.4 AUTOCORRELATION SUBTRACTION METHOD (AS)

This is a very simple as well as a rather crude procedure to obtain consistent LPC parameters. In this procedure autocorrelation of the noise is subtracted from the computed correlation of noisy speech. Then the correlation equation (2.2.6) is applied to obtain the parameters.

It has been mentioned in Section 1.2 that, in general, utterances consist of speech as well as silence. In this procedure, every speech frame goes through a speech/silence detection algorithm based on an energy threshold logic. The autocorrelation of every frame is computed. From the computed autocorrelation of the silence frames one gets the estimated autocorrelation of the noise. During the LPC analysis of a noisy speech frame, one may write:

$$\hat{r}_{ss}(j) = r_{yy}(j) - r_{vv}(j), \qquad (3.2.20)$$

which may be substituted in equation (2.2.6) to obtain estimates of the

LPC parameters.

As the frame length approaches infinity, the computed correlations converge to their true values. As a result:

$$\lim_{L\to\infty} \hat{r}_{ss}(j) = r_{ss}(j),$$

and therefore the estimates of the LPC parameters by this method will converge to their true values showing the cosistency of the estimator.

In the special case when the additive noise is white, equation (3.2.20) reduces to:

$$\hat{r}_{ss}(j) = r_{yy}(j) - L\,\sigma^2 \quad ; \text{ if } j=0 \qquad (3.2.21)$$

$$= r_{yy}(j) \qquad ; \text{ otherwise}$$

where σ^2 is the variance of the noise.

3.2.5 *SHIFTED YULE-WALKER EQUATIONS (SYW)*

It is mentioned in Section 3.2.1 that in general, the noise bandwidth is much wider than the speech bandwidth. This implies that the noise autocorrelation $r_{vv}(j)$ will approach zero faster than $r_{ss}(j)$ as j increases. Therefore for sufficiently large j, one may write:

$$\hat{r}_{yy}(j) = r_{ss}(j) + r_{vv}(j) \simeq r_{ss}(j). \qquad (3.2.22)$$

In the shifted Yule-Walker estimation of the parameters, the following equation is used:

$$r_{yy}(j) = \sum_{i=1}^{p} a_i r_{yy}(j-i), \qquad (3.2.23)$$

which is identical to equation (2.2.6), but in this case j assumes the values $p+\tau, \tau+1, \ldots, 2p+\tau-1$, where τ is sufficiently high such that $r_{vv}(\tau)$ may be neglected and equation (3.2.22) becomes true.

In the special case when the additive noise is white, $\tau=1$ may be used. A computationally efficient algorithm for the solution of the LPCs from shifted Yule-Walker equations is given in Appendix C.

With the assumption that τ is sufficiently high to make equation (3.2.22) true, it may be argued that as L approaches infinity, the computed autocorrelation function of the noisy speech will converge to the true autocorrelation function of the clean speech and therefore the solution of (3.2.23) will converge to the true values. Therefore the estimator is consistent.

3.3 LPC ESTIMATION THROUGH SPEECH ENHANCEMENT

An alternate approach to LPC parameter estimation for noisy speech is to use speech enhancement algorithms. Such an approach has been used by Shridhar [79]. It is expected that the use of an enhancement filter will reduce the noise level and the application of the LS estimator to the enhanced speech may yield nonsignificant bias.

However, it must be emphasized here that the enhanced speech will never be equal to the true speech and the algorithms will, therefore, yield neither unbiased nor consistent estimates [71]. Nevertheless in speech application, the data length (frame size) is too small, and unbiasedness and consistency may not be sufficient for our purpose. Low variances of the estimates are important as well. The usefulness of various speech enhancement algorithms is difficult to judge analytically. We will therefore verify their usefulness through practical implementation.

3.3.1 ADAPTIVE NOISE CANCELLATION PROCEDURE (ANC)

A technique for speech enhancement has been proposed [80] that utilizes the adaptive noise cancellation (ANC) approach. This method tries to find the best estimates (in the least squares sense) of the parameters of the clean speech from those of the noisy speech.

It has been mentioned in Section 3.2.3, that the speech samples delayed by a pitch period are highly correlated with the clean speech but nearly uncorrelated with the measurement noise. The ANC approach of speech enhancement takes advantage of this behaviour.

Let $\hat{s}_i$ be the estimate of the ith. sample of clean speech corresponding to the given noisy signal. Then the ANC procedure is based upon the following relation:

$$\hat{s}_i = \sum_{j=0}^{p} \alpha_j y_{i-T-j}. \tag{3.3.1}$$

In equation (3.3.1), p+1 is the order of prediction and T is the pitch period of the part of speech under consideration. It should be noted that this procedure is based upon the quasi-periodic nature of speech and only voiced sections of speech exhibit such a behaviour. Hence, in this approach, the unvoiced parts are either to be ignored or given a different treatment.

As described in an earlier section, speech has a dynamic behaviour that causes its parameters to change over time, though rather slowly. Pitch period, like other aspects of speech, is a function of time. Hence to keep the stationarity conditions, the given speech should be divided into frames of appropriate length L and pitch period be calculated for every voiced frame. This involves extra effort to take a voiced/unvoiced (v/uv) decision about each frame. An algorithm is presented in Appendix D for this purpose.

Once the speech is divided into frames the pitch period can be assumed constant in each frame. Note that for equations (3.3.1) to be valid, the frame length should be greater than the pitch period. This condition is also required for good estimates of the parameters. Hence, in the present case, this is not an extra restriction.

Multiplying equation (3.3.1) by y_{i-T-k} and taking expected value yields:

$$r_{ys}(T+k) = \sum_{j=0}^{p} \alpha_j r_{yy}(j-k), \qquad k=0,1,..,p \quad (3.3.2)$$

where r_{yy} and r_{ys} are respectively the autocorrelation function of $\{y_k\}$ and its crosscorrelation function with $\{s_k\}$. But it has already been mentioned that, because of its wide bandwidth, a noise sample is nearly uncorrelated to another noise sample a pitch period away. Hence:

$$r_{ys}(T+k) = r_{yy}(T+k),$$

and equation (3.3.2) becomes:

$$\sum_{j=0}^{p} \alpha_j r_{yy}(j-k) = r_{yy}(T+k), \qquad k=0,1,..,p. \quad (3.3.3)$$

Equation (3.3.3) represents a set of p+1 linear equations in the p+1 unknowns $\{\alpha_j\}$. Simultaneous solution of these equations yields the ANC estimates of the filter coefficients.

The next step is to apply the conventional LS method to the enhanced speech. However, it is not necessary to explicitly obtain enhanced speech. Instead, the approach followed by Ahmed [81] may be used to obtain the correlation functions of the filtered signal directly from those of the noisy speech. This direct approach increases computational efficiency as the time consuming process of correlation function estimation is not to be repeated for the enhanced signal. The steps for this procedure are now presented.

For each speech frame:

1. Calculate the autocorrelation functions $\{r_{yy}(j):j=0,1,..,2p\}$ of the observable noisy speech.
2. Compute the pitch period T. Take v/uv decision (see Appendix D). If the frame is unvoiced, go to step 1 with the next frame.
3. Solve equations (3.3.3) for $\{\alpha_i:i=0,1,..,p\}$. An efficient solution algorithm appears in Appendix E.

4. Compute the autocorrelation functions $\{r_{ss}(j):j=0,1,..,p\}$ of the estimated enhanced speech using:

$$r_{ss}(j) = B(q^{-1})r_{yy}(j), \qquad (3.3.4)$$

where $B(q^{-1}) = b_0 + \sum_{i=1}^{p} b_i(q^{i} + q^{-i}),$

and $b_i = \sum_{l=0}^{p-i} \alpha_l \alpha_{l+i}.$

Note that $r_{yy}(-j) = r_{yy}(j).$

5. Get the estimates of LPC parameters of the enhanced speech using equation (2.2.6) with the autocorrelation functions obtained in step 4.

3.3.2 LINEAR PREDICTIVE SMOOTHING (LPS)

In this enhancement technique, the clean speech signal is predicted as a linear combination of the past noisy samples, i.e.:

$$\hat{s}_i = \sum_{j=0}^{p} \beta_j y_{i-1-j}.$$

Minimization of the prediction error leads to the equations:

$$\sum_{j=0}^{p} \beta_j r_{yy}(j-k) = r_{yy}(k+1), \qquad k=0,1,..,p. \qquad (3.3.5)$$

which may be solved to obtain the filter parameters $\{\beta_j : j=0,1,..,p\}$. Note that these equations are similar to equations (3.3.3) with T=1. Hence the same solution procedure (given in Appendix E) may be used for their solution.

Once the parameters β_j have been obtained, a procedure similar to the one used with the ANC algorithm (Section 3.3.1) may be utilized for finding the estimates of autocorrelation functions of enhanced speech. In this approach v/uv decision and pitch period are not required. The procedure is applied to every frame.

3.3.3 *ADAPTIVE FILTERING TECHNIQUE (AFT)*

The adaptive filtering technique is based on the principle of orthogonality [82]. This principle states that ***the mean squared prediction error is minimized when the prediction error is uncorrelated with the input samples.*** It is to be noted that the AF algorithm requires some noise statistics to be known.

Let ε_i denote the error in predicting clean speech sample s_i from noisy speech $\{y_j\}$ using a (p+1)th. order estimation scheme. Then:

$$s_i = \sum_{j=0}^{p} \gamma_j y_{i-j} + \varepsilon_i .$$

Multiplying both sides of this equation by y_{i-k} and taking expected value gives:

$$r_{ys}(k) = \sum_{j=0}^{p} \gamma_j r_{yy}(j-k) + r_{y\varepsilon}(k). \qquad k=0,1,..p \quad (3.3.6)$$

Now using the above stated principle of orthogonality, minimization of the mean squared prediction error requires $\{y_i\}$ and $\{\varepsilon_i\}$ to be uncorrelated, i.e.:

$$r_{y\varepsilon}(k) = 0, \quad \text{for every } k.$$

Therefore, equation (3.3.4) becomes:

$$\sum_{j=0}^{p} \gamma_j r_{yy}(j-k) = r_{ys}(k). \qquad k=0,1,..p \quad (3.3.7)$$

But $r_{ys}=r_{ss}$ and from the independence of s_i and v_i:

$$r_{ss}(j) = r_{yy}(j) - r_{vv}(j). \qquad (3.3.8)$$

In addition if v_i is white, then:

$$r_{vv}(k) = L\,\sigma_v^2 \qquad \text{if } k=0, \qquad (3.3.9)$$

$$= 0 \qquad \text{otherwise.}$$

where σ_v^2 is used to denote the variance of the noise. Therefore equation (3.3.7) with (3.3.8) and (3.3.9) implies that:

$$\sum_{j=0}^{p} \gamma_j r_{yy}(j-k) = r_{yy}(0) - L\,\sigma_v^2, \text{ if } k=0 \qquad (3.3.10)$$

$$= r_{yy}(k) \qquad \text{otherwise.}$$

These p+1 equations for k=0,1,..,p can be solved by matrix inversion to get the values of the p+1 unknowns $\{\gamma_j\}$. The Toeplitz structure of these equations, however, allows for more efficient algorithms. It may be noted that equations (3.3.3) yield equations (3.3.10) when T is set equal to zero and $r_{yy}(0)$ is replaced by $r_{yy}(0)-L\sigma_v^2$ in only the right hand side. Hence the same algorithm (Appendix E) may be used with slight modifications to solve equations (3.3.10).

After estimating the filter coefficients γ_i, the procedure for estimating the enhanced speech is the same as the one outlined in Section 3.3.1 with α_i replaced by γ_i.

3.4 SPEAKER RECOGNITION

Once the estimates of the LPC parameters have been obtained by either of the methods discussed in Sections 3.2 and 3.3, the recognition procedure is the same as described in Section 2.4 for clean speech. The reference patterns and covariance matrices are obtained from clean speech using the conventional LPC estimation technique and stored. When a noisy test sample is recieved, the LPCs for each frame are estimated through one of the estimation techniques described above, the orthogonal LPCs are obtained through the speaker's reference transformation matrix. and distance is calculated with respect to the reference OLPCs.

3.5 *SUMMARY*

In this chapter, it has been shown that the estimates of the LPC parameters become biased and inconsistent when the conventional LS approach is used with noisy speech. To overcome this problem a number of procedures have been proposed. These procedures are divided into two groups.

In the first group, a set of consistent estimators is presented. They are GLS, ELS, IV, AS, and SYW methods.

The second group contains procedures that utilize a rather indirect, two steps approach. In the first step, these procedures enhance the noisy speech by various methods including ANC, LPS and AF techniques. The second step is the estimation of LPC parameters from the enhanced speech, by the conventional procedure.

Chapter IV

EXPERIMENTS AND RESULTS

4.1 CREATION OF THE DATABASE

The database for this study consisted of speech samples collected from eleven male speakers, all of approximately the same accent. Ten repetitions of each of the following two statements uttered by each of the speakers were recorded using high quality[1] recording equipment. The two statements were:

1. I was stunned by the beauty of the view, and
2. Our yacht slid around the point into the bay.

The recording was carried out in a normal room environment. At the next stage, the utterances were band pass filtered for a range of 300Hz. to 3kHz. so as to eliminate the noise due to recording equipment and recording environment. The filtered speech was then digitized at 7.9 kHz. using an 8 bit A/D converter installed on an APPLE//e microcomputer. The digitized data files were, then, transferred to the IBM3033 computer where the actual processing of speech samples was done. The SSP routine GAUSS was used to add various levels of zero mean, normally distributed random noise to the clean speech.

To create reference templates of each speaker, the first five noise-free utterances of statement number 1 were used. Each utterance was divided into frames of length 200 samples (corresponding to about 25.3

---

[1] For specification of the instruments, see Appendix F.

milli-seconds at a sampling frequency of 7.9 kHz.). Each frame was then tested for silence and the silent frames appearing at the beginning and end of each utterance were eliminated. This decision was taken on the basis of the frame energy. The algorithm presented in Appendix A was used to solve (2.2.6) with p=12 in order to get the LPC parameters in each frame. The dimensionality of the sample space was further reduced by extracting orthogonal LPC parameters. For this purpose, covariance matrices of the LPC parameters in each of the reference utterances were calculated using (2.4.1). The unique covariance matrix for each speaker was then found as the weighted average given by (2.4.3) of the matrices obtained in (2.4.1). The program EIGEN of the Scientific Subroutines Package (SSP) was used to find the eigenvalues and eigenvectors of the unique covariance matrix. The OLPC parameters were then obtained using equation (2.4.4) and were averaged over all the reference utterances of each speaker using (2.4.5). The eigenvalues, eigenvectors and the OLPC parameters were then stored as the reference templates.

4.2 RECOGNITION WITH CONVENTIONAL LPC ESTIMATION

The remaining five utterances of each speaker were used as test utterances assuming the speaker to be unknown. Each utterance was divided into frames of length 200 samples and the silent frames appearing at the beginning and end were eliminated as described above. The conventional LS procedure was applied to extract the LPC parameters as described in Section 4.1. These LPC parameters were then used to calculate OLPCs using the eigenvectors stored in the reference template of each known speaker. The distance measure d_m given by equation (2.4.6) was obtained for each reference speaker. The unknown speaker was identified as the reference speaker corresponding to the minimum value of d_m.

The algorithm was tested using all twelve, the last nine and the last six LPC parameters by setting q=1, q=4 and q=7 respectively in equation (2.4.6).

At the next stage, the SSP routine GAUSS was used to add zero mean, normally distributed random noise at NTS[2] ratios of .05 and .10. The noise was added to the test utterance samples and the above mentioned recognition procedure was repeated. The results[3] of these experiments are summarized in Table 1.

It may be observed from these results that in the noise free case the conventional approach yields exceptionally good results. The accuracy was not changed even when the number of parameters was decreased to 6.

It has been shown in Section 3.1 that the least squares estimates of the LPC parameters are biased when noise is present in speech. The degradation in recognition accuracy in presence of noise was therefore quite expected. Table 1 shows this degradation. Moreover it may also be inferred that the recognition accuracy will further decrease if more noise is added.

---

[2] Noise to signal (NTS) ratio is defined as the ratio of standard deviation of noise to signal standard deviation.

[3] Recognition accuracy is defined as percentage of the number of correct decisions with respect to the total number of decisions.

TABLE 1

RECOGNITION ACCURACIES USING THE CONVENTIONAL APPROACH

NTS	Number of Parameters		
	12	9	6
0	100%	98.18%	100%
.05	67.28%	65.46%	61.82%
.10	45.45%	41.81%	41.81%

Table 2 shows an example of the calculated distances for the noise-free case.

TABLE 2

Statement number: 6 (type A)

Approach: Conventional

NTS ratio: 0

Number of parameters: 12

REFERENCE

TEST	ABBASI	MUFTI	AKHLAQ	JAWED	AHMED	SHABBAR	MATLOOB	TAEMOOR	ZAIDI	RASHEED	HAFEEZ
ABBASI	12.27	183.98	527.79	109.14	432.99	413.29	643.73	523.85	270.30	220.45	678.46
MUFTI	247.81	11.62	614.24	109.69	379.00	659.44	842.09	287.99	351.74	70.27	536.28
AKHLAQ	615.95	390.94	18.62	325.19	333.95	345.51	445.84	330.44	332.22	240.91	220.83
JAWED	338.55	483.18	337.38	65.92	294.53	209.73	167.21	287.16	354.01	323.06	304.91
AHMED	296.63	283.12	1034.30	182.56	18.19	446.53	312.23	154.42	449.26	308.24	485.17
SHABBAR	392.17	1055.66	1155.86	381.43	722.76	11.63	303.93	945.71	907.79	713.23	759.17
MATLOOB	403.77	694.50	503.19	241.57	318.10	148.77	33.99	423.99	553.63	437.65	406.26
TAEMOOR	466.22	149.58	661.29	222.72	282.70	661.38	518.14	9.51	374.83	143.42	384.51
ZAIDI	402.34	206.36	240.83	278.56	456.56	402.38	265.84	373.64	28.28	194.22	468.56
RASHEED	817.96	180.61	674.22	405.40	802.77	969.07	1435.71	360.45	687.54	16.24	732.75
HAFEEZ	1952.84	3089.00	2486.48	1484.56	1922.49	1243.66	1306.06	1853.13	4074.77	1245.08	21.42

4.3 RECOGNITION WITH IV, AS AND SYW ESTIMATION OF LPCS

When it was confirmed that the recognition potential of the conventional LPC estimates are severely degraded in presence of noise, the algorithms discussed in Sections 3.2 and 3.3 were applied. It should be clear that the reference templates and recognition procedure for all these techniques were identical to those for the conventional approach, which were extracted from the noise-free utterances.

The GLS and ELS techniques described in Section 3.2 have been found to yield good estimates of parameters in transfer function model identification but in estimation of autoregressive parameters, they were found appropriate only for low order models. It had been observed that in case of high order models these methods yield very poor estimates of autoregressive parameters. It has been mentioned in Section 3.2 that these are iterative relaxation procedures used to minimize strongly nonlinear functions. The problem in such cases, however, is that there may exist more than one minima and the algorithm may converge to a local minimum rather than the global minimum.

In our application, estimation of the parameters of a 12th. order LPC model was required. Experiments have shown that using GLS and ELS procedures to such high order models almost always results in convergence to a local minimum, which gives rise to misleading results. Specifically, it has been observed that for such models the estimates obtained by GLS and ELS procedures are no better than those obtained by the LS method. On these grounds, no attempt was made to apply these procedures to speaker recognition.

In order to estimate the LPC parameters using the IV approach, pitch period was required to be known. After dividing the test utterance into frames and chopping off the start and end silent frames, each frame was subjected to the v/uv test. The procedure adopted for this purpose is given in Appendix D. A frame decided as unvoiced was ignored. For the voiced frames, pitch period was estimated using the method given in Appendix D. The algorithm of Appendix B was then used to solve equation (3.2.19) in order to obtain the IV estimates of the LPC parameters. It should be noted that in the algorithm of Appendix B, two estimates of these parameters have been obtained. It has been suggested there that a combination of these two be used to get low variance estimates of the LPC parameters. An average of these two sets of estimates was used and the results at various noise levels are presented in Table 3.

In the autocorrelation subtraction approach, equation (3.2.21) was used to estimate the autocorrelation functions of clean speech. The algorithm of Appendix A was used to solve equation (2.2.6) with the autocorrelation functions of the clean speech replaced by their estimates. The estimates of the LPC parameters obtained as the solution of (2.2.6) were then used in the recognition process. The results are summarized in Table 4.

The process of LPC estimation using SYW equations requires the solution of (3.2.23). The value of τ was set at $p+1$ (=13 in our case) and the algorithm of Appendix C was used to solve equations (3.2.23). The reason for choosing $\tau=p+1$ was to exclude $r_{yy}(0)$ from (3.2.23). Since the measurement noise was assumed white, any nonzero value of j

TABLE 3

RECOGNITION ACCURACIES WITH THE IV PROCEDURE

	Number of Parameters		
NTS	12	9	6
0	29.09%	21.82%	25.45%
.05	20.00%	18.18%	16.36%
.10	29.09%	20.00%	20.00%

will satisfy (3.2.22). Table 5 contains the result of using the SYW estimates for recognition.

TABLE 4

RECOGNITION ACCURACIES WITH THE AS METHOD

NTS	Number of Parameters		
	12	9	6
0	100%	98.18%	100%
.05	27.27%	27.27%	23.63%
.10	20.00%	16.36%	14.54%

TABLE 5

RECOGNITION ACCURACIES USING THE SYW EQUATIONS

NTS	Number of Parameters		
	12	9	6
0	16.36%	12.73%	14.54%
.05	12.73%	16.36%	10.91%
.10	18.18%	14.54%	18.18%

4.4 RECOGNITION WITH ANC, LPS AND AFT ENHANCEMENT

In case of the speech enhancement techniques, various algorithms were used to enhance the noisy speech. The conventional OLPC technique (see Section 4.2) was then used to extract the OLPC parameters from the enhanced speech. These parameters were then tested against the reference templates formed using the clean speech (Section 4.1). It has already been mentioned in Chapter 3 that in the proposed algorithms, it is not necessary to explicitly estimate the enhanced speech. Rather, estimates of the autocorrelation functions of the enhanced speech can be obtained from those of the clean speech directly. The conventional LPC estimation technique is then applied to the enhanced speech autocorrelation functions.

In the ANC approach of speech enhancement, a v/uv decision was taken for each frame. Pitch periods of voiced frames were estimated (Appendix D), and the enhancement filter coefficients were computed from equation (3.3.3) in conjunction with the algorithm in Appendix E. Then the autocorrelation functions of the enhanced speech were computed using equation (3.3.4). For the unvoiced frames, the filter coefficients of the preceding voiced frame were used in equation (3.3.4). The enhanced autocorrelation functions were then subjected to conventional LPC analysis and consequent distance calculation. The results using this approach are given in Table 6.

TABLE 6

RESULTS USING THE ANC ALGORITHM

	Number of Parameters		
NTS	12	9	6
0	56.36%	47.27%	43.63%
.05	63.64%	49.09%	45.45%
.10	45.45%	40.00%	38.18%

The adaptive filtering technique requires noise variance to be known. In practice, the noise variance may be estimated from the silent frames of speech. For our study, however, the noise variance was assumed to be known. The enhancement filter coefficients were obtained by the solution of (3.3.8). The algorithm given in Appendix E was used with the necessary modifications. Clean speech autocorrelation functions and its LPC parameters were then estimated using the approach of Section 3.3.1.

To limit the fluctuation of filter coefficients from frame to frame, a smoothing algorithm of the following type was used:

$$(\gamma_i)_{k+1} = \xi(\gamma_i)_k + (1-\xi)(\gamma_i)_{calculated} , \qquad (4.4.1)$$

where $(\gamma_i)_k$ is the ith. filter coefficient of the kth. frame. $(\gamma_i)_{calculated}$ is the ith. filter coefficient of the kth. frame obtained from equation (3.3.3), and ξ is an appropriate factor. It was thought that since γ_i's were calculated from a relatively short data which yields rather high variance estimates of the coefficients, use of such a filter may reduce the fluctuation in the estimates leading to relatively low variance smooth estimates. Various values of ξ were tried and the best results were obtained with a fraction of 0.5. These results are presented in Table 7.

The linear predictive smoothing technique of speech enhancement requires the solution of the autocorrelation equations (3.3.5). In order to implement the LPS technique, enhancement filter coefficients were found for each frame (after chopping the beginning and end silence parts) using equations (3.3.5). The algorithm given in Appendix E

TABLE 7

RECOGNITION ACCURACIES WITH THE AFT

NTS	Number of Parameters		
	12	9	6
0	100%	98.18%	100%
.05	98.18%	96.36%	90.91%
.10	80.00%	72.73%	72.73%

(with T=1) was used for their solution. A similar filter coefficient smoothing scheme as used with AFT was applied in order to smooth the filter coefficients. Equations (3.3.4) were then applied to obtain the estimates of the clean speech autocorrelation functions. These estimates were then used in the conventional procedure of LPC estimation. The recognition procedure was then applied to these LPCs. The results of this experiment appear in Table 8.

TABLE 8

RECOGNITION ACCURACIES WITH THE LPS TECHNIQUE

NTS	Number of Parameters		
	12	9	6
0	34.54%	29.09%	25.45%
.05	30.91%	25.45%	20.00%
.10	32.72%	20.00%	12.73%

4.5 DISCUSSION OF THE RESULTS

The results of the various speaker recognition algorithms discussed in the previous section are summarized in Table 9, for the case when 12 LPC parameters were used.

It may be observed from this table that the AS and SYW methods prove to be total failures. The results of the IV procedure are also not good. It has been shown in Section 3.2 that these procedures yield consistent estimates. To take advantage of consistency property a large value of L is required. However, in our case L was only 200, which is small when compared to the large number of parameters (12 in our case) to be estimated. Hence the variance of estimation was too high. Another reason for the failure of these methods was that the coefficient matrices

TABLE 9

SUMMARY OF RECOGNITION ACCURACIES

ALGORITHM	NTS RATIO		
	0	.05	.10
CONV.	100%	67.28%	45.45%
IV	29.09%	20.00%	29.09%
AS	100%	27.27%	20.00%
SYW	16.36%	12.73%	18.18%
ANC	56.36%	63.64%	45.45%
AFT	100%	98.18%	80.00%
LPS	34.54%	30.91%	32.72%

for the solution of LPC parameters were ill conditioned. Such a situation leads to erroneous estimation of LPCs.

The procedures based on speech enhancement yielded relatively better results. One of these, the AFT, showed the best overall performance. Although these procedures can not produce unbiased or consistent estimates (Section 3.3), the better results indicate that probably the bias in the estimates were very small. It should be noted that the coefficient matrices appearing in the solutions of the autocorrelation equations of this class of procedures are covariance matrices of physically existing signals. Such matrices with zero lag autocorrelation functions on their diagonals are diagonally dominant and nonsingular. It appears that the diagonal dominance and consequent nonsingularity of the coefficient matrices may be the reason for the expected small bias. Which may, in turn, be the reason for better performance.

4.6 TEXT INDEPENDENT RECOGNITION USING AFT

In order to measure the text independent recognition potential of the AFT, statement 2 given in Section 4.1 was used as test utterance against reference templates based upon statement 1. Two recognition schemes, the conventional and the AFT, were tried. Statement 2 as uttered by one of the speakers was longer than could be digitized using the available microcomputer. Therefore only ten test speakers could be considered. The results of this study are presented in Tables 10 and 11.

A comparison of these results shows that, in noisy case, the AFT is superior to the conventional technique as text independent speaker discriminator. It should be mentioned here that text independent recognition, in general, requires longer utterances as compared to text dependent recognition. However, with the limited memory (64K)

TABLE 10

TEXT INDEPENDENT RECOGNITION USING CONVENTIONAL METHOD

NTS	Number of Parameters		
	12	9	6
0	100%	90%	70%
.05	60%	60%	30%
.10	30%	30%	20%

available on the Apple//e computer, it was not possible to digitize utterances of more than 3.2 seconds duration. It is expected that longer utterances will yield further improved results.

TABLE 11

TEXT INDEPENDENT RECOGNITION USING AFT

NTS	Number of Parameters		
	12	9	6
0	100%	90%	70%
.05	80%	80%	50%
.10	60%	50%	10%

4.7 SUMMARY

This chapter describes the speaker recognition procedures implemented in this study and the results obtained from them. The first section discusses the various aspects of the creation of database and the procedure employed to form the reference templates. The second section deals with the results obtained when the conventional LPC estimation procedure was used both in noise free and noisy cases. The next section is devoted to the presentation of the results of various algorithms employed to improve the recognition performance. Then these results have been discussed and the last section consists of a comparison of the conventional technique and AFT as text independent speaker discriminators.

Throughout this chapter, it has been tried to adopt such an approach that the reader may get guidance if he wishes to perform these experiments himself. He is often referred to a number of appendices which include efficient algorithms that can be employed during the implementation of these experiments.

Chapter V

CONCLUDING REMARKS

5.1 SUMMARY AND CONCLUSIONS

It has been observed that the LS method of LPC parameter estimation is successful for speaker recognition in the noise free case. But in the noisy case, the parameter estimates become highly biased, resulting in severe degradation to the recognition ability of the LPC procedure. The primary goal of this study was to find which technique of LPC parameter estimation could be used in the noisy environment giving higher recognition accuracy. Among the various approaches considered, it was found that the estimation of LPCs through AFT speech enhancement is well capable of recognizing correct speaker in noisy environment.

In addition to successfully achieving the primary goal, the following contributions were made:

1. The technique of using OLPC parameters for speaker recognition was implemented and tested. The claims about the high recognition accuracy of this procedure in noise free case were confirmed. The algorithm was also tried with noisy case and the observations about its severe degradation were found to comply with what has been mentioned in the related literature.
2. A number of various techniques based upon modified LS methods and speech enhancement procedures were tried to get better

results in the noisy case. The modified LS methods were IV, AS and SYW procedures. These procedures were found to yield poor recognition accuracies. This situation was because of two reasons: i) although these methods should yield consistent estimates but consistency requires large frame length which in our case was only 200 samples, and ii) the coefficient matrices for LPC estimation were ill conditioned due to loss of diagonal dominance.

The second set of procedures were based upon speech enhancement before LPC estimation. Three enhancement techniques, namely ANC, LPS and AFT were considered. An algorithm for obtaining autocorrelation functions of the enhanced speech without explicit enhancement was used. These procedures, based on speech enhancement could not yield unbiased or consistent estimates. But empirically it was found that these procedures yielded better results as compared to those obtained by modified LS methods. This indicated that probably the biases in the estimates were very small.

A comparative study of these procedures showed that the recognition procedure based on the adaptive filtering technique (AFT) of speech enhancement has a much higher speaker discrimination potential in noisy environment as compared to the conventional technique.

3. The AF technique and the conventional approach were tried as text independent speaker recognizers. A comparison of the results showed the superiority of the AF technique over the

conventional one in the noisy case. It was, however, felt that the full text independent speaker recognition potential of these techniques could not be achieved due to insufficient utterance length. It is expected that the performance of the AF technique will further improve if the utterance length is increased.

4. In order to make the solution of certain special classes of autocorrelation equations computationally feasible, efficient algorithms were developed. These algorithms appear in Appendices B, C and E.

5.2 *RECOMMENDATIONS FOR FUTURE RESEARCH*

In light of the preliminary studies carried out for this work and from the experience gained, it is felt that possibilities of research in the following fields exist:

1. A number of speech characteristics other than LPCs like parcor coefficients, formant frequencies, formant amplitudes etc. may be investigated for the case of speaker recognition in noisy environment.
2. The OLPC parameters may be applied to other areas of speech processing like speech recognition etc.
3. The effect of additive colored noise and multiplicative noise on speaker and speech recognition potential of various algorithms is another interesting and important area of further research.
4. The various LPC estimators discussed in this thesis may be implemented for noisy speech recognition and a comparative study on their performance may be carried out.

5. The text independent recognition capabilities of the algorithms given in the thesis may be studied in detail using larger utterances.

REFERENCES

1. B. S. Atal, "Automatic Recognition of Speakers from Their Voices," Proc. IEEE, vol. 64, no. 4, Apr. 1976.
2. R. S. Cheung and B. S. Einstein, "Feature Selection via Dynamic Programming for Text Independent Speaker Identification," IEEE Trans. on Acoust., Speech, Signal Processing, vol. ASSP-26, no. 5, Oct. 1978.
3. S. Furui, "Comparison of Speaker Recognition Methods Using Statistical Features and Dynamic Features," IEEE Trans. On Acoust., Speech, Signal Processing, vol. ASSP-29, no. 3, Jun. 1981.
4. A. E. Rosenberg, "Automatic Speaker Verification: A Review," Proc. IEEE, vol. 64, no. 4, Apr. 1976.
5. J. J. Wolf, "Efficient Acoustic Parameters for Speaker Recognition," J. Acoust. Soc. Amer., vol. 51, pt. 2, pp. 2044-2055, June 1972.
6. M. R. Sambur, "Selection of Acoustic Features for Speaker Identification," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, pp. 176-182, Apr. 1975.

7. S. Pruzansky and M. V. Mathews, "Talker-Recognition Procedure Based on Analysis of Variance," J. Acoust. Soc. Amer., vol. 36, pp. 2041-2047, Nov. 1964.

8. W. S. Mohn Jr., "Two Statistical Feature Evaluation Techniques Applied to Speaker Identification," IEEE Trans. Comput., vol. C-20, pp. 979-987, Sept. 1971.

9. P. DeSouza and P. J. Thompson, "LPC Distance Measures and Statistical Tests with Paricular Reference to the Likelihood Ratio," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-30, no. 2, Apr. 1982.

10. A. E. Rosenberg and M. R. Sambur, "New Techniques for Automatic Speaker Verification," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, no. 2, Apr. 1975.

11. J. M. Tribolet, L. R. Rabiner and M. M. Sondhi, "Statistical Properties of an LPC Distance Measure," IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-27, No. 5, Oct. 1979.

12. G. R. Doddington, "A Method of Speaker Verification," Ph. D. dissertation, Univ. Wisconsin, 1970.

13. J. D. Markel and A. H. Gray Jr., "Linear Prediction of Speech," Springer-Verlag, Berlin, Heidelberg, New York.

14. J. L. Flanagan, "Computers that Talk and Listen," Proc. IEEE, Apr. 1976.

15. L. G. Kersta, "Voiceprint Identification," Nature, vol. 196, 1962.

16. F. R. Clarke and R. W. Becker, "Comparison of Techniques for Discriminating Among Talkers," J. Speech & Hearing Res., vol. 12, 1969.

17. F. R. Clarke, R. W. Becker, and J. C. Nixon, "Characteristics that Determine Speaker Recognition," Air Force Systems Command Tech. Rept. No. RSD-TR-66-636, 1966.

18. K. N. Stevens, et. al., "Speaker Authentification and Identification: A Comparison of Spectrographic and Auditory Presentations of Speech Material," J. Acoust. Soc. Amer., vol. 44, 1968.

19. M. A. Young and R. A. Campbell, "Effects of Context on Talker Identification," J. Acoust. Soc. Amer., vol. 42, 1967.

20. J. R. Carbonell, et. al., "Speaker Authentification Techniques," Bolt Beranek Newman Rept. No. 1296, Cambridge, Mass., 1965.

21. F. McGehee, "The Reliability of the Identification of Human Voice," J. Gen. Psycol., vol. 17, 1937.

22. I. Pollack, J. M. Pickett, and W. H. Sumby, "On the Identification of Speakers by Voice," J. Acoust. Soc. Amer., vol. 26, 1954.

23. W. D. Voiers, "Perceptual Bases of Speaker Identity," J. Acoust. Soc. Amer., vol. 36, 1964.

24. W. Hargreaves and J. A. Starkweather, "Recognition of Speaker Identity," Lang. Speech, vol. 6, pp. 63-67, 1963.

25. S. Furui, F. Itakura and S. Saito, "Talker Recognition by Long-Time Averaged Speech Spectrum," Electron Comm. Jap., vol. 55-A, pp. 54-61, 1972.

26. K. P. Li and G. W. Hughes, "Talker Differences as They Appear in Correlation Matrices of Continous Speech Spectra," J. Acoust. Soc. Amer., vol. 55, pp. 833-837, 1974.

27. B. S. Atal, "Effectiveness of Linear Prediction Characteristics of the Speech Waves for Automatic Speaker Identification and Verification," J. Acoust. Soc. Amer., vol. 55, pp. 1304-1312, 1974.

28. S. K. Das and W. S. Mohn, "A Scheme for Speech Processing in Automatic Speaker Verification," IEEE Trans. Audio Electroacoust., vol. AU-19, pp. 32-43, Mar. 1971.

29. J. W. Glen and N. Kleiner, "Speaker Identification Based on Nasal Phonation," J. Acoust. Soc. Amer., vol. 43, pp. 368-372, 1968.

30. L. S. Su, K. P. Li and K. S. Fu, "Identification of Speakers by Use of Nasal Coarticulation," J. Acoust. Soc. Amer., vol. 56, pp. 1876-1882, 1974.

31. R. C. Lummis, "Speaker Verification by Computer Using Speech Intensity for Temporal Registration," IEEE Trans. Audio Electroacoust., vol. AU-21, pp. 80-89, 1973.

32. B. S. Atal, "Automatic Speaker Recognition Based on Pitch Contours," J. Acoust. Soc. Amer., vol. 52, pp. 1687-1697, Dec. 1972.

33. G. R. Doddington, "Speaker Verification - Final Report," Rome Air Developement Center, Griffis AFB, NY, Tech. Rep. RADC 74-179, Apr. 1974.

34. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, pp. 67-72, 1975.

35. A. E. Rosenberg, "Evaluation of an Automatic Speaker Verification System Over Telephone Lines," Bell Syst. Tech. J., 1976.

36. L. R. Rabiner, A. E. Rosenberg and S. E. Levinson, "Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition," IEEE Trans. on Acoust., Speech, Signal Processing, vol. ASSP-26, no. 6, Dec. 1978.

37. H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," IEEE Trans. on Acoust., Speech, Signal Processing, vol. ASSP-26, pp. 43-49, Feb. 1978.

38. A. E. Rosenberg, "Listener Performance in Speaker Verification Tasks," IEEE Trans. Audio Electroacoust., vol. AU-21, pp. 221-225, 1973.

39. R. W. Schafer and L. R. Rabiner, "Digital Representation of Speech Signals," Proc. IEEE, vol. 63, pp. 662-677, Apr. 1975.

40. S. Pruzansky, "Pattern-Matching Procedure for Automatic Talker Recognition," J. Acoust. Soc. Amer., vol. 35, pp. 354-358, Mar. 1963.

41. P. D. Bricker, R. Gnanadesikan, M. V. Mathews, S. Pruzansky, P. A. Tukey, K. W. Wachter and J. L. Warner, "Statistical Techniques for Talker Identification," Bell Syst. Tech. J., vol. 50, pp. 1427-1454, Apr. 1971.

42. K. P. Li, J. E. Dammann and W. D. Chapman, "Experimental Studies in Speaker Verification Using an Adaptive System," J. Acoust. Soc. Amer., vol. 40, pp. 966-978, 1966.

43. G. S. Ramishvili, "Experiments on Automatic Verification of Speakers," Proc. 2nd Int. Joint Conf. Pattern Recognition, Copenhagen, Denmark, pp. 389-393, 1974.

44. J. E. Luck, "Automatic Speaker Verification Using Cepstral Measurements," J. Acoust. Soc. Amer., vol. 46, pp. 1026-1031, 1969.

45. M. R. Sambur, "Speaker Recognition Using Orthogonal Linear Prediction," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-26, no. 24, Aug. 1976.

46. R. E. Wohlford, E. H. Wrench and B. P. Landell, "A Comparison of Four Techniques for Automatic Speaker Recognition," Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, Denver, Colorado, 1980, pp. 908-911.

47. G. E. P. Box and G. M. Jenkins, "Time Series Analysis Forecasting and Control," San Francisco, Calif.: Holden-Day, 1970.

48. G. D. Hair and R. W. Rekieta, "Automatic Speaker Verification Using Phoneme Spectra," J. Acoust. Soc. Amer., vol. 51, p. 131(A), 1972.

49. G. D. Hair and R. W. Rekieta, "Mimic Resistance of Speaker Verification Using Phoneme Spectra," J. Acoust. Soc. Amer., vol. 51, p. 131(A), 1972.

50. E. Bunge, "Automatic Speaker Recognition by Computers," Proc. Carnahan Conf. Crime Countermeasures, 1975.

51. M. S. Ahmed, "Estimating the Parameters of a Noisy AR-Process by using a Bootstrap Parameter," Proc. Int. Conf. on ASSP, Paris, 1982, vol. 1, pp. 152-155.

52. G. R. Doddington, "A Method of Speaker Verification," J. Acoust. Soc. Amer., vol. 49, pt. 1, p. 139(A), Jan. 1971.

53. B. S. Atal, "Automatic Speaker Recognition Based on Pitch Contours," Ph. D. dissertation, Polytech. Inst. Brooklyn, Brooklyn, NY., June 1968.

54. A. M. Noll, "Cepstrum Pitch Determination," J. Acoust. Soc. Amer., vol. 41, pp. 293-309, Feb. 1967.

55. M. M. Sondhi, "New Methods of Pitch Detection," IEEE Trans. Audio Electroacoust., vol. AU-16, pp. 262-266, June 1968.

56. B. Gold and L. R. Rabiner, "Parallel Processing Techniques for Estimating pitch periods of Speech in the Time Domain," J. Acoust. Soc. Amer., vol. 46, pp. 442-449, Aug. 1969.

57. J. D. Markel, "The SIFT Algorithm for Fundamental Frequency Estimation," IEEE Trans. Audio Electroacoust., vol. AU-20, pp. 367-377, Dec. 1972.

58. P. Garvin and Ladefoged, "Speaker Identification and Message Identification in Speech Recognition," Phonetics, vol. 9, no. 4, pp. 193-199, 1963.

59. B. S. Atal and S. L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," J. Acoust. Soc. Amer., vol. 50, pt. 2, pp. 637-655, Aug. 1971.

60. F. Itakura and S. Saito, "An Analysis-Synthesis Telephony System Based on Maximum Likelihood Method," Electron. Comm. Japan, vol. 53A, pp. 36-43, 1970.

61. J. Makhoul, "Linear Prediction: A Tutorial review," Proc. IEEE, vol. 63, pp. 561-580, Apr. 1975.

62. J. D. Markel, A. H. Gray, Jr., and H. Wakita, "Linear Prediction of Speech: Theory and Practice," Speech Communications Res. Lab., Santa Barbara, CA, SCRL Monogr. 10, Sept. 1973.

63. R. W. Schafer and L. R. Rabiner, "System for Automatic Formant Analysis of Voiced Speech," J. Acoust. Soc. Amer., vol. 47, pp. 634-648, Feb. 1970.

64. J. Olive, "Automatic Formant Tracking by a Newton-Raphson Technique," J. Acoust. Soc. Amer., vol. 50, pt. 2, pp. 661-670, Aug. 1971.

65. J. D. Markel, "Digital Inverse Filtering-A New Tool for Formant Trajectory Estimation," IEEE Trans. Audio Electroacoust., vol. AU-20, pp. 129-137, June 1972.

66. S. S. McCandless, "An Algorithm for Automatic Formant Extraction Using Linear Prediction Spectra," IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-22, pp. 135-141, Apr. 1974.

67. G. R. Doddington, J. L. Flanagan and R. C. Lummis,"Automatic Speaker Verification by Non-linear Time Alignment of Acoustic Parameters," U. S. Patent 3, 700, 815, issued Oct. 24, 1972.

68. M. R. Sambur, "Speaker Recognition and Verification Usin Linear Prediction Analysis," Ph. D. dissertation, Dep. Elec. Engg., Mass. Inst. Technol., Sept. 1972. also M. I. T. Res. Lab., Electron. Quart. Prog. Rep. 108, pp. 261-268, Jan. 1973.

69. M. R. Sambur, "An Efficient Linear Prediction Vocoder," Bell Syst. Tech. J., Dec. 1975.

70. T. C. Hsia, "System Identification," Lexington Books, Toronto.

71. M. S. Ahmed, "Rapid Parameter Estimation Algorithms using the ELS Principle," Systems and Control Letters 2(1982)209-216, North Holland Publishing Company.

72. M. S. Ahmed, "Three Extended Correlation Algorithms for LPC Analysis of Noisy Data," Unpublished.

73. L. Ljung, T. Soderstrom and I. Gustavvson, "Counter Examples to General Convergence of a commonly used Recursive Identification Algorithm," IEEE Trans. Auto. Control, vol. AC-20, pp. 643-652, 1975.

74. M. S. Ahmed, "Fast IV Algorithms for System Identification," Proc. IEEE Conf. on Circuits and Systems, Newportbeech, Calif., 1983.

75. J. D. Wise, J. R. Caprio and T. W. Parks, "Maximum Likelihood Pitch Estimation," IEEE Trans. on Acoust., Speech, Signal Processing, vol. ASSP-24, no. 5, Oct. 1976.

76. M. J. Ross, H. L. Shaffer, A. Cohen, R. Freudberg and H. J. Manley, "Average Magnitude Difference Function Pitch Extractor," . IEEE Trans. on Acoust., Speech, Signal Processing, vol. ASSP-22, no. 5, Oct. 1974.

77. J. J. Dubnowski, R. W. Schaffer and L. R. Rabiner, "Real-Time Digital Hardware Pitch Detector," IEEE Trans. on Acoust., Speech, Signal Processing, vol. ASSP-24, no. 1, Feb. 1976.

78. L. R. Rabiner, M. J. Cheng and C. A. McGenegal, "A Comparative Performance Study of Several Pitch Detection Algorithms," IEEE Trans. on Acoust., Speech, Signal Processing, vol. ASSP-24, no. 5, Oct. 1976.

79. M. Baraniecki and M. Shridhar, "A Speaker Verification Algorithm for Speech Utterances Corrupted by Noise with Unknown Statistics." Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, vol. 1, pp. 904-907, Denver, Colorado, 1980.

80. M. R. Sambur, "Adaptive Noise Cancelling for Speech Signals," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-26, no. 5, Oct. 1978.

81. M. S. Ahmed, "Fast GLS Algorithm for Parameter Estimation," Automatica, vol. 20, no. 2, pp. 231-236, 1984.

82. A. Popoulis, "Probability, Random Variables and Stochastic Processes," McGraw-Hill Book Company, New York, 1965.

83. E. H. Wrench Jr., "A Real Time Implementation of a Text Independent Speaker Recognition System," Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, Atlanta, Georgia, 1980, pp. 908-911.

Appendix A

A RECURSIVE ALGORITHM FOR LPC ESTIMATION

Consider the autocorrelation version of equations for LPC estimation given by (2.2.6) as:

$$\sum_{i=1}^{p} r_{ss}(i-k)a_i = r_{ss}(k). \qquad k=1,2,..,p \quad (A.1)$$

In order to develop the recursion formulae, replace a_j by a_{pj} in equations (A.1). The first subscript represents the order of linear prediction. Moreover, define the normalized autocorrelation function $\{\rho_j\}$ by:

$$\rho_j = r_{ss}(j) \;/\; r_{ss}(0).$$

Hence, equations (A.1) reduce to:

$$\sum_{i=1}^{p} \rho_{i-k}\, a_{pi} = \rho_k. \qquad k=1,2,..p \quad (A.2)$$

Note that $\rho_0 = 1$, $\rho_{-i} = \rho_i$ and $|\rho_i| \le 1$, where $|.|$ denotes Euclidean norm. Denote the system of equations (A.2) of order p by Λ_p and define $R_p = [\rho_{i-j}]_{p \times p}$. Then the solution of Λ_2 is given by:

$$\begin{pmatrix} a_{21} \\ \\ a_{22} \end{pmatrix} = R_2^{-1} \begin{pmatrix} \rho_1 \\ \\ \rho_2 \end{pmatrix} \tag{A.3}$$

Next consider Λ_3:

$$\begin{aligned} a_{31} + \rho_1 a_{32} + \rho_2 a_{33} &= \rho_1 \\ \rho_1 a_{31} + \quad a_{32} + \rho_1 a_{33} &= \rho_2 \\ \rho_2 a_{31} + \rho_1 a_{32} + \quad a_{33} &= \rho_3 \end{aligned} \tag{A.4}$$

From the first two equations of (A.4), a_{31} and a_{32} may be written in terms of a_{33} as:

$$\begin{pmatrix} a_{31} \\ \\ a_{32} \end{pmatrix} = R_2^{-1} \begin{pmatrix} \rho_1 \\ \\ \rho_2 \end{pmatrix} - R_2^{-1} \begin{pmatrix} \rho_2 \\ \\ \rho_1 \end{pmatrix} a_{33}$$

$$= \begin{pmatrix} a_{21} \\ \\ a_{22} \end{pmatrix} - \begin{pmatrix} a_{22} \\ \\ a_{21} \end{pmatrix} a_{33} \qquad \text{from (A.3)}$$

Thus:

$$a_{3i} = a_{2i} - a_{2,3-i} a_{33}. \qquad i=1,2. \tag{A.5}$$

Substituting values of a_{31} and a_{32} in the third equation of (A.4):

$$a_{33} = (\rho_3 - \sum_{j=1}^{2} a_{2j}\rho_{3-j}) \;/\; (1- \sum_{j=1}^{2} a_{2j}\rho_j) \qquad \text{(A.6)}$$

Generalization of (A.5) and (A.6) yields the following recursion farmulae for solving Λ_{l+1} from the solution of Λ_l :

$$a_{l+1,l+1} = (\rho_{l+1} - \sum_{j=1}^{l} a_{lj}\, \rho_{l+1-j}) \;/\; (1 - \sum_{j=1}^{l} a_{lj}\, \rho_j)$$

and,

$$a_{l+1,i} = a_{li} - a_{l+1,l+1}\, a_{l,l-i+1}, \qquad i=1,2,..,l.$$

The algorithm starts with the solution of Λ_1 given by $a_{11}=\rho_1$. After p-1 recursions, the solution of the original system is obtained.

It can be easily shown that the above algorithm requires $(3p^2+p-2)/2$ multiplications and divisions and $(p^2+3p-4)/2$ subtractions for the solution of a pth. order system. This includes the p divisions required to find ρ_i from r(i). For the solution of a 12th. order system, this amounts to 221 multiplications and divisions as compared to more than 576 for the general methods of solution. Hence, for large order systems, a substantial saving in computation time results. This is very important because the solution of Λ_p is to be performed for every frame and even a short utterance, of say 3 seconds duration, may consist of 80 to 100 frames.

Appendix B

SOLUTION OF SYSTEM OF EQUATIONS IN IV METHOD

Consider equations (3.2.19) for LPC estimation by IV method:

$$\sum_{i=1}^{p} r_{yy}(T+j-i)a_i = r_{yy}(T+j). \qquad j=1,2,..,p \quad (B.1)$$

Define normalized autocorrelation functions as:

$$\upsilon_i = r_{yy}(i) \;/\; r_{yy}(T).$$

Then equations (B.1) may be written as:

$$\sum_{i=1}^{p} \upsilon_{T+j-i}a_{pi} = \upsilon_{T+j}, \qquad j=1,2,..,p \quad (B.2)$$

where a_i has been replaced by a_{pi}. Note that $\upsilon_T=1$. Call the system given by (B.2) as $\mathcal{T}_p$. Define matrix Y_k by $Y_k=[\upsilon_{T+k+1-i-j}]_{k\times k}$.

$\mathcal{T}_4$ may be written as:

$$\begin{aligned}
\upsilon_{T+3}a_{41} + \upsilon_{T+2}a_{42} + \upsilon_{T+1}a_{43} + a_{44} &= \upsilon_{T+4} \\
\upsilon_{T+2}a_{41} + \upsilon_{T+1}a_{42} + a_{43} + \upsilon_{T-1}a_{44} &= \upsilon_{T+3} \\
\upsilon_{T+1}a_{41} + a_{42} + \upsilon_{T-1}a_{43} + \upsilon_{T-2}a_{44} &= \upsilon_{T+2} \\
a_{41} + \upsilon_{T-1}a_{42} + \upsilon_{T-2}a_{43} + \upsilon_{T-3}a_{44} &= \upsilon_{T+1}
\end{aligned} \qquad (B.3)$$

The last three equations of (B.3) give:

$$\begin{pmatrix} a_{41} \\ a_{42} \\ a_{43} \end{pmatrix} = Y_3^{-1} \begin{pmatrix} \upsilon_{T+3} \\ \upsilon_{T+2} \\ \upsilon_{T+1} \end{pmatrix} - Y_3^{-1} \begin{pmatrix} \upsilon_{T-1} \\ \upsilon_{T-2} \\ \upsilon_{T-3} \end{pmatrix} a_{44} \tag{B.4}$$

But the first term on the right hand side of (B.4) is the solution of T_3. Assume that $(b_{33}\ b_{32}\ b_{31})^T$ is such that:

$$\begin{aligned} \upsilon_{T+2}b_{33} + \upsilon_{T+1}b_{32} + b_{31} &= \upsilon_{T-1} \\ \upsilon_{T+1}b_{33} + b_{32} + \upsilon_{T-1}b_{31} &= \upsilon_{T-2} \\ b_{33} + \upsilon_{T-1}b_{32} + \upsilon_{T-2}b_{31} &= \upsilon_{T-3} \end{aligned} \tag{B.5}$$

Hence (B.4) becomes:

$$\begin{pmatrix} a_{41} \\ a_{42} \\ a_{43} \end{pmatrix} = \begin{pmatrix} a_{31} \\ a_{32} \\ a_{33} \end{pmatrix} - \begin{pmatrix} b_{33} \\ b_{32} \\ b_{31} \end{pmatrix} a_{44},$$

or

$$a_{4i} = a_{3i} - b_{3,4-i}a_{44}. \qquad i=1,2,3 \tag{B.6}$$

Substitution of these values of a_{41}, a_{42} and a_{43} in the first equation of (B.3) yields:

$$a_{44} = (\upsilon_{T+4} - \sum_{j=1}^{3} \upsilon_{T+j}a_{3,4-j}) \;/\; (1 - \sum_{j=1}^{3} \upsilon_{T+j}b_{3j}). \qquad (B.7)$$

Generalizations of (B.7) and (B.6) are:

$$a_{k+1,k+1} = (\upsilon_{T+k+1} - \sum_{j=1}^{k} \upsilon_{T+j}a_{k,k+1-j})/(1-\sum_{j=1}^{k} \upsilon_{T+j}b_{kj}) \qquad (B.8)$$

$$a_{k+1,i} = a_{ki} - b_{k,k+1-i}\, a_{k+1,k+1}. \qquad i=1,2,..,k \quad (B.9)$$

Rearranging equations (B.5) gives:

$$\begin{aligned} \upsilon_{T-2}b_{31} + \upsilon_{T-1}b_{32} + b_{33} &= \upsilon_{T-3} \\ \upsilon_{T-1}b_{31} + b_{32} + \upsilon_{T+1}b_{33} &= \upsilon_{T-2} \\ b_{31} + \upsilon_{T+1}b_{32} + \upsilon_{T+2}b_{33} &= \upsilon_{T-1} \end{aligned} \qquad (B.10)$$

A similar approach as used to obtain equations (B.8) and (B.9) yields the following general solution for (B.10):

$$b_{k+1,k+1} = (\upsilon_{T-k-1} - \sum_{j=1}^{k} \upsilon_{T-j}b_{k,k+1-j})/(1-\sum_{j=1}^{k} \upsilon_{T-j}a_{kj}) \qquad (B.11)$$

$$b_{k+1,i} = b_{ki} - a_{k,k+1-i}\, b_{k+1,k+1}. \qquad i=1,2,..,k \quad (B.12)$$

The recursive algorithm for the solution of (B.1) may, therefore, be organized as:

1. Set $a_{11}=\upsilon_{T+1}$, $b_{11}=\upsilon_{T-1}$ and k=1.
2. Use (B.8) and then (B.9) to find $\{a_{k+1,i}: i=1,2,..,k+1\}$.

3. If $k+1=p$ then stop.

4. Use (B.11) and then (B.12) to get $\{b_{k+1,i}: i=1,2,..,k+1\}$.

5. Set $k:=k+1$ and go to step 2.

It should be emphasized here that the biproduct parameters $\{b_{pj}\}$ obtained in step 4 above can be used to obtain better estimates of the LPC parameters. These parameters are, in fact, the result of backward LPC estimation procedure. But as speech is assumed to be stationary in a frame, the true values of both these sets of parameters $\{a_{pj}\}$ and $\{b_{pj}\}$ are the same. Hence the parameters obtained as the average of these two sets will be relatively lower variance estimates of the LPC coefficients.

Appendix C

SOLUTION OF SHIFTED YULE-WALKER EQUATIONS

Consider the SYW equations as given by (3.2.23):

$$\sum_{i=1}^{p} r_{yy}(j-i)a_i = r_{yy}(j). \qquad j=\tau,\tau+1,..,\tau+p-1 \quad (C.1)$$

Define the kth. order system Ω_k by:

$$\sum_{i=1}^{k} \omega_{j-i}a_{ki} = \omega_j, \qquad j=\tau,\tau+1,..,\tau+p-1 \quad (C.2)$$

where ω_i has been defined as $\omega_i=r_{yy}(i)/r_{yy}(\tau-1)$ and a_i has been replaced by a_{ki}. Note that $\omega_{\tau-1}=1$.

Define the matrix W_k by $W_k=[\omega_{\tau+k-i-j}]_{k\times k}$.

A rearrangement of the equations of Ω_3 gives its solution as:

$$\begin{pmatrix} a_{31} \\ a_{32} \\ a_{33} \end{pmatrix} = W_3^{-1} \begin{pmatrix} \omega_{\tau+2} \\ \omega_{\tau+1} \\ \omega_{\tau} \end{pmatrix} \qquad (C.3)$$

Consider the equations in Ω_4 written as:

$$\omega_{\tau+2}a_{41} + \omega_{\tau+1}a_{42} + \omega_{\tau}\;a_{43} + \quad a_{44} = \omega_{\tau+3}$$

$$\omega_{\tau+1}a_{41} + \omega_{\tau}\;a_{42} + \quad a_{43} + \omega_{\tau-2}a_{44} = \omega_{\tau+2}$$

$$\omega_{\tau}\;a_{41} + \quad a_{42} + \omega_{\tau-2}a_{43} + \omega_{\tau-3}a_{44} = \omega_{\tau+1} \tag{C.4}$$

$$a_{41} + \omega_{\tau-2}a_{42} + \omega_{\tau-3}a_{43} + \omega_{\tau-4}a_{44} = \omega_{\tau}$$

The last three equations yield:

$$\begin{pmatrix} a_{41} \\ a_{42} \\ a_{43} \end{pmatrix} = W_3^{-1} \begin{pmatrix} \omega_{\tau+2} \\ \omega_{\tau+1} \\ \omega_{\tau} \end{pmatrix} - W_3^{-1} \begin{pmatrix} \omega_{\tau-2} \\ \omega_{\tau-3} \\ \omega_{\tau-4} \end{pmatrix} a_{44} \tag{C.5}$$

Let $\{b_{3i}: i=1,2,3\}$ be such that:

$$W_3 \begin{pmatrix} b_{33} \\ b_{32} \\ b_{31} \end{pmatrix} = \begin{pmatrix} \omega_{\tau-2} \\ \omega_{\tau-3} \\ \omega_{\tau-4} \end{pmatrix} \tag{C.6}$$

Using these values in the second term on the right hand side of (C.5) and replacing the first term by (C.3), one gets:

$$\begin{pmatrix} a_{41} \\ a_{42} \\ a_{43} \end{pmatrix} = \begin{pmatrix} a_{31} \\ a_{32} \\ a_{33} \end{pmatrix} - \begin{pmatrix} b_{33} \\ b_{32} \\ b_{31} \end{pmatrix} a_{44},$$

or

$$a_{4i} = a_{3i} - b_{3,4-i}a_{44}. \qquad i=1,2,3 \quad (C.7)$$

Solution of the first equation of (C.4) for a_{44} with (C.7) gives:

$$a_{44} = (\omega_{\tau+3} - \sum_{j=1}^{3} \omega_{\tau-1+j}a_{3,4-j}) \;/\; (1 - \sum_{j=1}^{3} \omega_{\tau-1+j}b_{3j}).$$

Generalizing this equation and (C.7) gives:

$$a_{k+1,k+1} = (\omega_{\tau+k} - \sum_{j=1}^{k} \omega_{\tau-1+j}a_{k,k+1-j})/(1 - \sum_{j=1}^{k} \omega_{\tau-1+j}b_{kj}) \qquad (C.8)$$

$$a_{k+1,i} = a_{ki} - b_{k,k+1-i}\, a_{k+1,k+1}, \qquad i=1,2,..,k \quad (C.9)$$

A similar treatment yields the following recursive formulae for the solution of (C.6):

$$b_{k+1,k+1} = (\omega_{\tau-k-2} - \sum_{j=1}^{k} \omega_{\tau-1-j}b_{k,k+1-j})/(1 - \sum_{j=1}^{k} \omega_{\tau-1-j}a_{kj}) \qquad (C.10)$$

$$b_{k+1,i} = b_{ki} - a_{k,k+1-i}b_{k+1,k+1}, \qquad i=1,2,..,k \quad (C.11)$$

Hence, the recursive algorithm for the solution of SYW equations (C.1) may be summarized as:

1. Set $a_{11}=\omega_\tau$, $b_{11}=\omega_{\tau-2}$ and $k=1$.
2. Solve (C.8) and (C.9) to obtain $\{a_{k+1,i}: i=1,2,..,k+1\}$.
3. If $k+1=p$ then stop.
4. Solve (C.10) and (C.11) to get $\{b_{k+1,i}: i=1,2,..,k+1\}$.
5. Set $k:=k+1$ and go to step 2.

Appendix D

PITCH DETECTION AND V/UV DECISION

The following algorithm for pitch period extraction has been proposed by Wise et.al. [75] and is based upon a maximum likelihood pitch estimate. This algorithm has been claimed to be resistant to additive white noise, which may also be extended to colored noise. In presence of realistic environmental noise, the performance has been shown to be promising [75]. A great advantage in using this algorithm is that it utilizes a parameter that can be used for v/uv decision making. Moreover, with a little extra effort, pitch period can be estimated with a resolution finer than one sample.

The outlines of the procedure are given here. Those interested in detail may refer to [75].

Let L be the length of the frame for which pitch estimation is desired. Let $r_{yy}(i)$ denote the autocorrelation function of the speech in that frame. Define a function g(P) by:

$$g(P) = 2P \sum_{i=1}^{N-1} r_{yy}(iP) \,/\, L,$$

where $N \equiv$ the greatest integer less than or equal to L/P.

Suppose $\hat{P}$ be that value of P which maximizes g(P). Mathematically speaking, let:

$$\max_P g(P) = g(\hat{P}).$$

Then, $\hat{P}$ is found to be the most satisfactory estimate of the pitch period provided that the frame contains voiced speech.

It may be noted that it is required to calculate $g(P)$ only for those values of P which may be possible candidates for pitch period. The maximum out of these values of $g(P)$ is chosen and the corresponding P is taken as the estimate of pitch period.

To obtain resolution finer than one sample, non-integer values of P are also to be considered. For this discussion, the reader is referred to [75].

VOICED/UNVOICED DECISION: The parameter $\hat{P}$ obtained in the foregoing discussion as the best estimate of pitch period is helpful in taking v/uv decision. The procedure for this decision is based upon the fact that in voiced frames, the energy due to the periodic portion of the signal constitutes a large portion of the total energy. It has been suggested [75] that the ratio $g(\hat{P})/r_{yy}(0)$ be utilized for this purpose. This implies that a frame be decided as voiced for which the value of $g(\hat{P})/r_{yy}(0)$ is greater than a certain threshold.

It was, however, found experimentally that a better decision may be taken if the deciding ratio is replaced by the quantity $\{\delta r_{yy}(\hat{P}) + g(\hat{P})\} / r_{yy}(0)$ where δ is a constant.

To summarize this concept, a frame was decided as voiced for which:

$$[\{\delta r_{yy}(\hat{P}) + g(\hat{P})\} / r_{yy}(0)] > t.$$

The values of the weighting factor δ and the threshold t used in this work were 3.0 and 2.0 respectively.

Appendix E

SOLUTION OF SYSTEM OF EQUATIONS IN ANC

Consider the sytem of equations (3.3.3):

$$\sum_{j=0}^{p} r_{yy}(j-k)\alpha_j = r_{yy}(T+k). \qquad k=0,1,..,p \quad (E.1)$$

In terms of ρ_j (defined in Appendix A) these equations become:

$$\sum_{j=0}^{p} \rho_{j-k}\alpha_{pj} = \rho_{T+k}, \qquad k=0,1,..,p \quad (E.2)$$

where α_j has been replaced by α_{pj}. Note that (E.2) is a p+1 th. order system. Let this system be denoted by Γ_p. Define matrix Q_k by $Q_k=[\rho_{k+2-i-j}]_{(k+1)\times(k+1)}$.

Consider the third order system Γ_2:

$$\rho_2\alpha_{20} + \rho_1\alpha_{21} + \alpha_{22} = \rho_{p+2}$$

$$\rho_1\alpha_{20} + \alpha_{21} + \rho_1\alpha_{22} = \rho_{p+1}$$

$$\alpha_{20} + \rho_1\alpha_{21} + \rho_2\alpha_{22} = \rho_p$$

The solution of this system is given by:

$$\begin{pmatrix} \alpha_{20} \\ \alpha_{21} \\ \alpha_{22} \end{pmatrix} = Q_2^{-1} \begin{pmatrix} \rho_{p+2} \\ \rho_{p+1} \\ \rho_p \end{pmatrix} \tag{E.3}$$

Next consider Γ_3:

$$\begin{aligned} \rho_3\alpha_{30} + \rho_2\alpha_{31} + \rho_1\alpha_{32} + \alpha_{33} &= \rho_{p+3} \\ \rho_2\alpha_{30} + \rho_1\alpha_{31} + \alpha_{32} + \rho_1\alpha_{33} &= \rho_{p+2} \\ \rho_1\alpha_{30} + \alpha_{31} + \rho_1\alpha_{32} + \rho_2\alpha_{33} &= \rho_{p+1} \\ \alpha_{30} + \rho_1\alpha_{31} + \rho_2\alpha_{32} + \rho_3\alpha_{33} &= \rho_p \end{aligned} \tag{E.4}$$

The last three equations of Γ_3 give:

$$\begin{aligned} \begin{pmatrix} \alpha_{30} \\ \alpha_{31} \\ \alpha_{32} \end{pmatrix} &= Q_2^{-1} \begin{pmatrix} \rho_{p+2} \\ \rho_{p+1} \\ \rho_p \end{pmatrix} - Q_2^{-1} \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{pmatrix} \alpha_{33} \\ &= \begin{pmatrix} \alpha_{20} \\ \alpha_{21} \\ \alpha_{22} \end{pmatrix} - \begin{pmatrix} \nu_{22} \\ \nu_{21} \\ \nu_{20} \end{pmatrix} \alpha_{33}. \end{aligned} \tag{E.5}$$

The first term on the right hand side of (E.5) has been obtained from (E.3) and $(\nu_{22}\ \nu_{21}\ \nu_{20})$ are such that:

$$\rho_2\nu_{20} + \rho_1\nu_{21} + \nu_{22} = \rho_3$$

$$\rho_1\nu_{20} + \nu_{21} + \rho_1\nu_{22} = \rho_2 \qquad \text{(E.6)}$$

$$\nu_{20} + \rho_1\nu_{21} + \rho_2\nu_{22} = \rho_1$$

Equations (E.5) now give:

$$\alpha_{3i} = \alpha_{2i} - \nu_{2,2-i}\,\alpha_{33}. \qquad i=0,1,2 \quad \text{(E.7)}$$

Using these values of $\alpha_{3,i}$ in the first equation of (E.4) gives:

$$\alpha_{33} = (\rho_{p+3} - \sum_{j=1}^{3} \rho_j\alpha_{2,3-j}) \,/\, (1 - \sum_{j=1}^{3} \rho_j\nu_{2,j-1}). \qquad \text{(E.8)}$$

Generalizing equations (E.8) and (E.7), one gets:

$$\alpha_{k+1,k+1} = (\rho_{p+k+1} - \sum_{j=1}^{k+1} \rho_j\alpha_{k,k+1-j})/(1 - \sum_{j=1}^{k+1} \rho_j\nu_{k,j-1}) \qquad \text{(E.9)}$$

$$\alpha_{k+1,i} = \alpha_{ki} - \nu_{k,k-i}\,\alpha_{k+1,k+1}. \qquad i=0,1,..,k \quad \text{(E.10)}$$

Comparing equations (E.6) with equations (A.4), it may easily be noted that ν_{ij} are conventional estimates of LPC parameters given by:

$$\nu_{k+1,k+1} = (\rho_{k+2} - \sum_{j=1}^{k+1} \rho_j\nu_{k,k+1-j})/(1 - \sum_{j=1}^{k+1} \rho_j\nu_{k,j-1}) \qquad \text{(E.11)}$$

$$\nu_{k+1,i} = \nu_{ki} - \nu_{k,k-i}\,\nu_{k+1,k+1}. \qquad i=0,1,..,k \quad \text{(E.12)}$$

The solution algorithm may thus be summarized as:

1. Set $\alpha_{00}=\rho_p$, $\nu_{00}=\rho_1$ and $k=0$.
2. From (E.9) and (E.10) find $\{\alpha_{k+1,i}:i=0,1,\ldots,k+1\}$.
3. If $k+1=p$ then stop.
4. Solve (E.11) and (E.12) to get $\{\nu_{k+1,i}:i=0,1,\ldots,k+1\}$.
5. Set $k:=k+1$ and go to step 2.

Appendix F

SPECIFICATION OF THE EQUIPMENTS

AMPLIFIER: EAI-TIMS (Available on the motherboard)

------ Based on LM301.

------ Gain: Max. 4.7.

(Two amplifiers were used in series, adjusted to give an output of from -5 to +5 volts.)

A/D CONVERTER: MOUNTAIN COMPUTER INC. (Installed on an Apple//e microcomputer)

------ Resolution rated at 8 bits (including sign bit).

------ Conversion method : Successive approximation.

------ Input impedance for AC : 1 kOhm.

------ Analog input voltage range : from -5 to +5 volts.

------ Channel conversion time : 9 μseconds.

------ Accuracy : Absolute = +/- 3% FSR.

Relative = +/- 1 LSB.

MICROPHONE: REVOX M 3500

------ Cardioid type.

------ Impedance : 200 Ohms.

------ Frequency response : 40 Hz. to 18 kHz.±3 dB.

TAPE RECORDER: REVOX B-77 STEREO TAPE RECORDER

------ Tape transport mechanism :

3-motor tape drive, 2 Ac driven spooling motors,

1 AC driven capstan motor,

(electronically regulated).

------ Tape speed : 7.5 ips.

------ Wow and flutter (as per DIN 45507) : < 0.08%

------ Tape slip : max. 0.2%

------ Tape transport control :

Integrated control logic with tape motion

sensor.

------ Equalization (as per NAB) :

50 μsec./3180 μsec.

------ Frequency response : 30 Hz. to 20 kHz.+2/-3 dB.

50 Hz. to 15 kHz.±1.5 dB.

------ Peak recording level :

514 nWb/m corresponds to 6 dB. above 0 VU.

------ Distortion : < 0.2% at 0 VU (257 nWb/m)

< 0.5% at 0 VU+6 dB. (514 nWb/m)

------ Signal to noise ratio : Better than 67 dB.

(weighted as per ASA-A, measured via tape)

------ Crosstalk (at 1000 Hz.) : Better than 60 dB.

(monophonic).

------ Input per channel MIC (unbalanced) :

0.15 mV/2,2 kOhms.

------ Output per channel : 1.55 V.

FILTER: GENERAL RADIO COMPANY 1952 UNIVERSAL FILTER

------ Frequency range :

Cut-off frequencies : Adjustable 4 Hz. to 60 kHz. in four ranges.

Pass-band limits : Low-frequency response to dc (approx 0.7 Hz. with ac input coupling) in Low Pass and Band Reject modes. High-frequency response uniform ±0.2 dB to 300 kHz. in High Pass and Band Reject modes.

------ Filters :

Filter characteristics : Filters: fourth order (four-pole) Chebyshev approximations to ideal magnitude response. The nominal pass-band ripple: ±0.1 dB (±0.2 dB max); nominal attenuation at the calibrated cut-off frequency: 3 dB; initial attenuation rate: 30 dB per octave.

Minimum bandwidth : 26% (approx 1/3 octave) in

Band Pass mode.

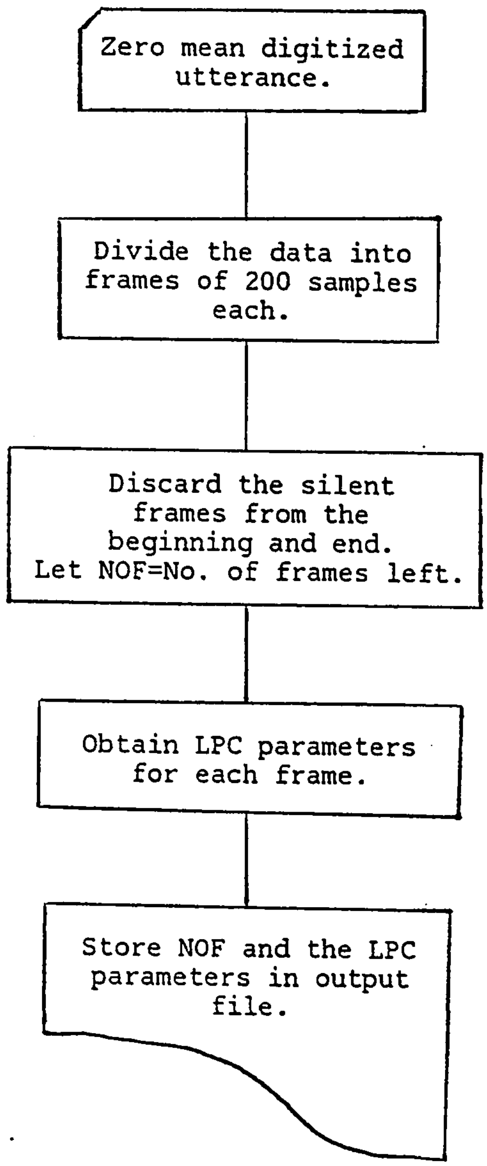
------ Noise : <100 μV in an effective bandwidth

of 50 kHz.

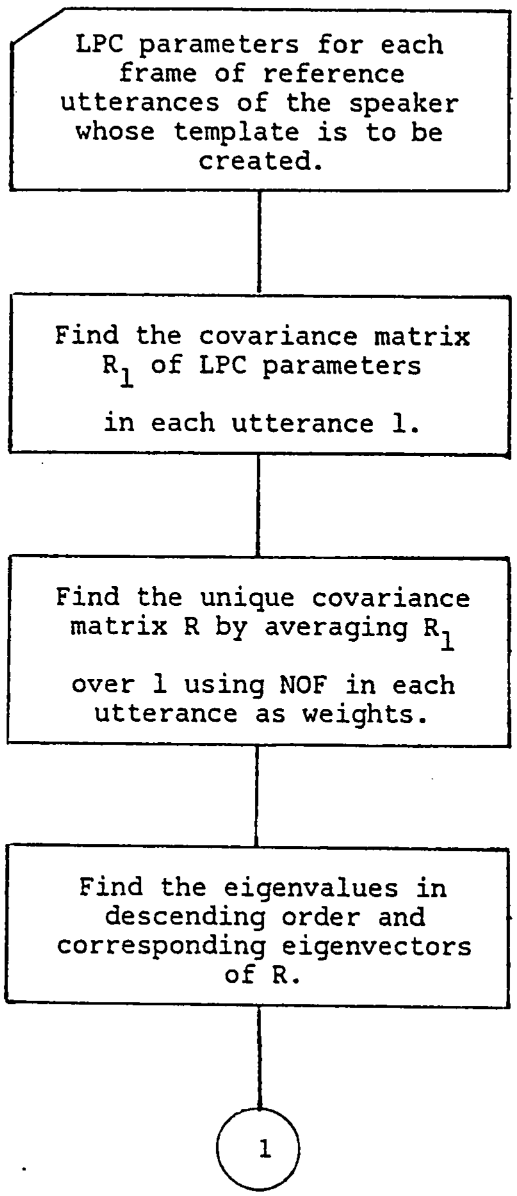
Appendix G

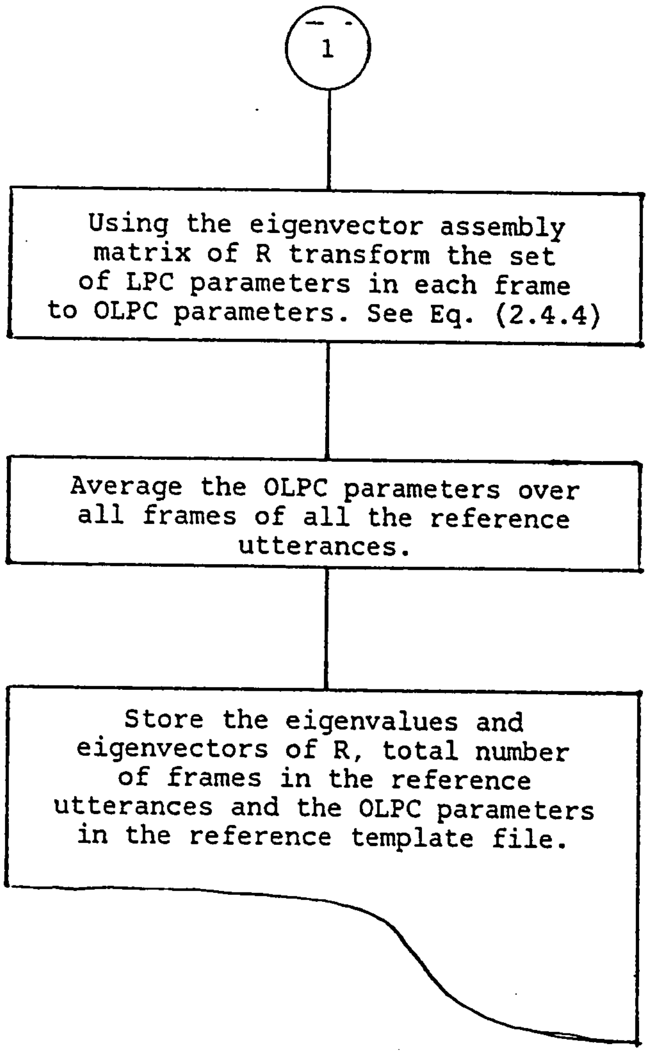
FLOWCHARTS

1. Flow Chart for LPC Estimation from an utterance using the Conventional method

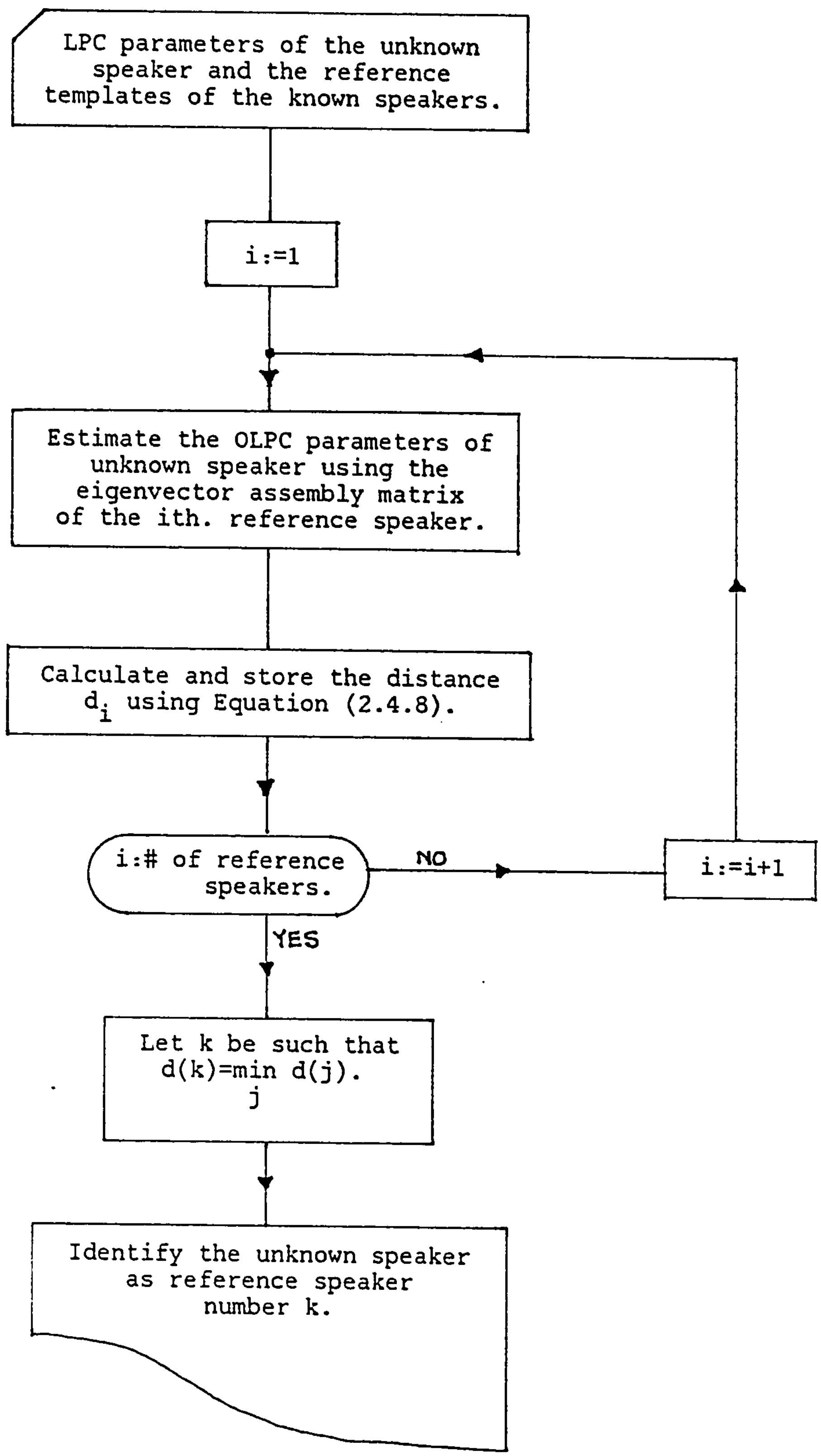


2. Flow Chart for Creation of Reference Templates

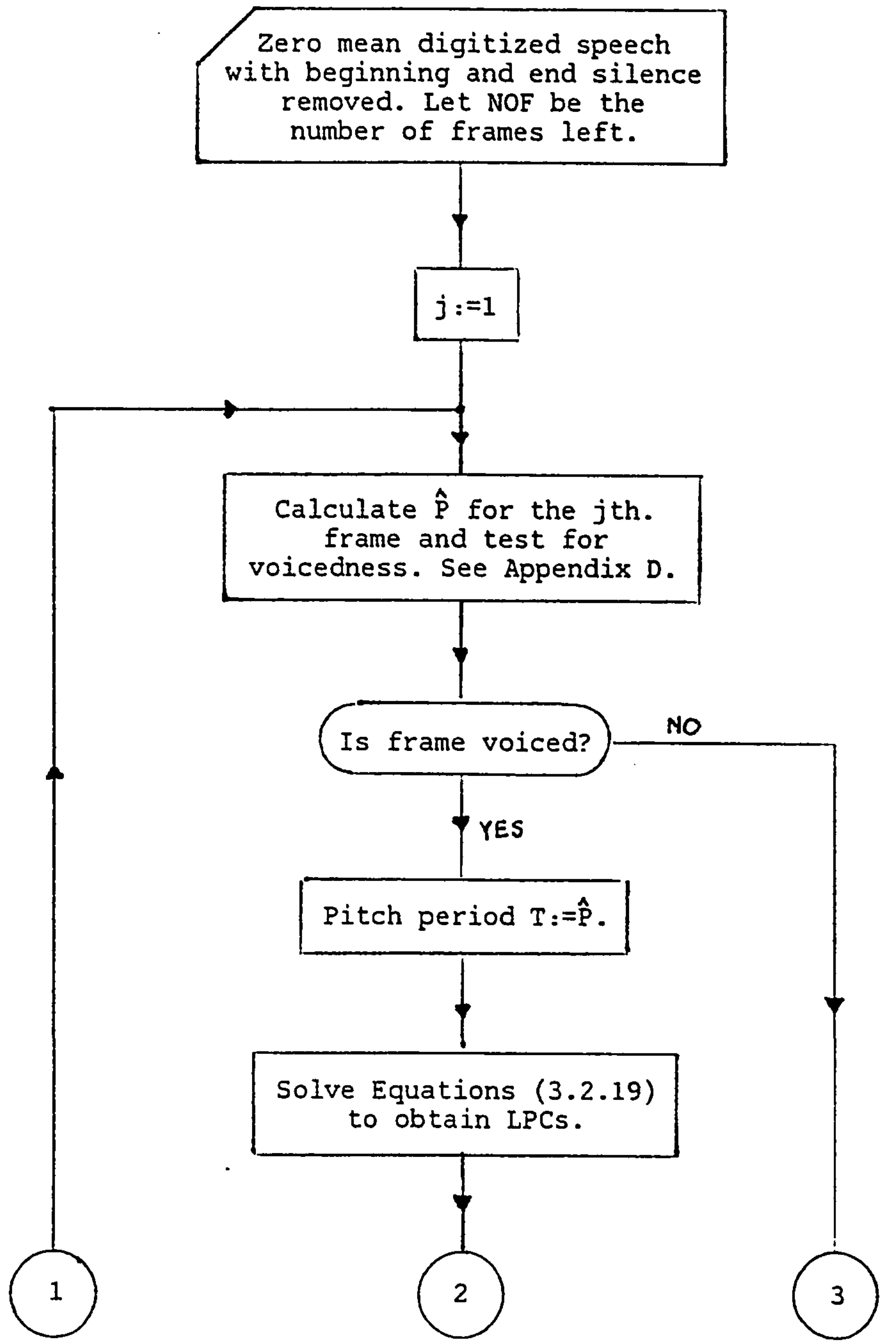


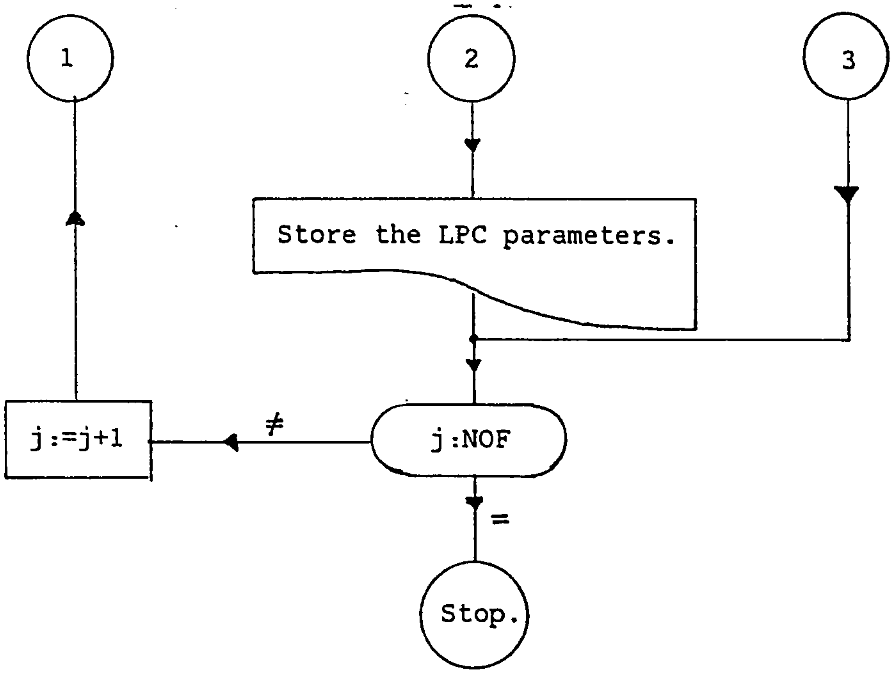

1
Using the eigenvector assembly matrix of R transform the set of LPC parameters in each frame to OLPC parameters. See Eq. (2.4.4)
Average the OLPC parameters over all frames of all the reference utterances.
Store the eigenvalues and eigenvectors of R, total number of frames in the reference utterances and the OLPC parameters in the reference template file.

3. The Recognition Procedure

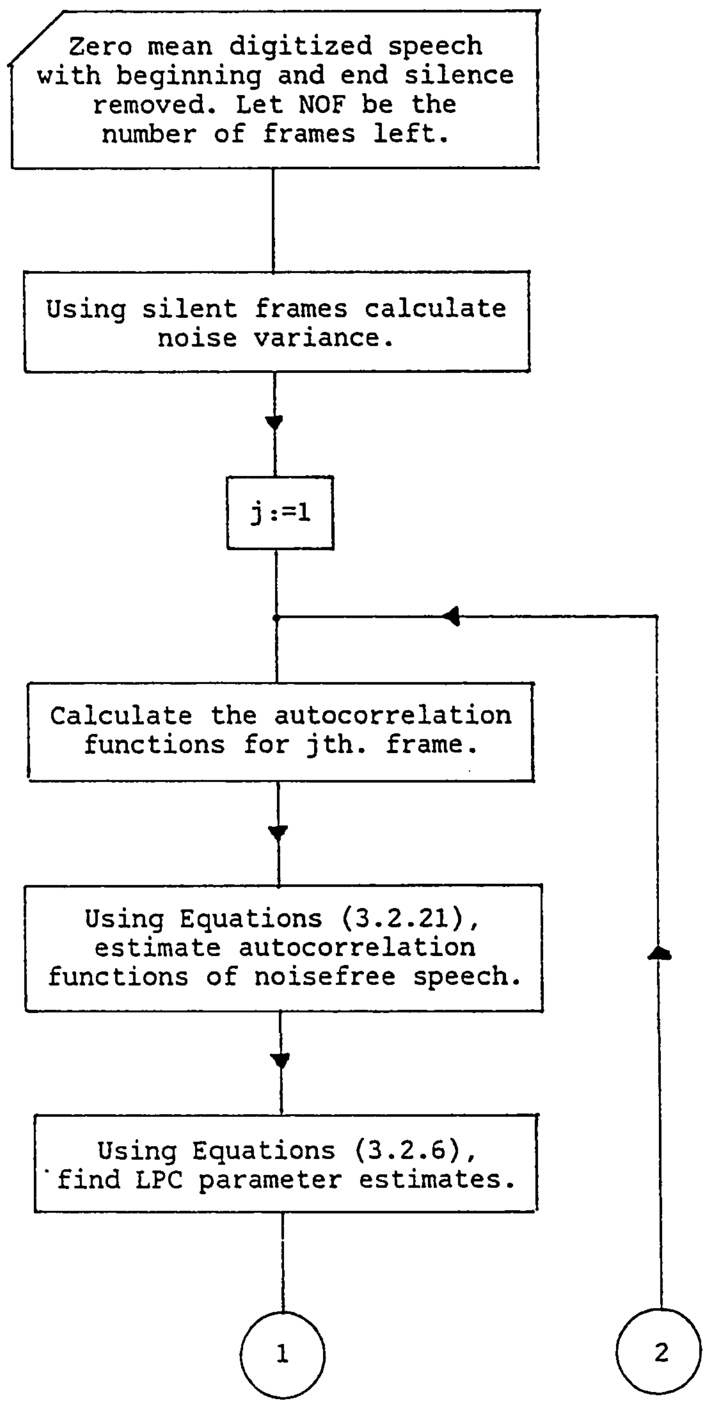


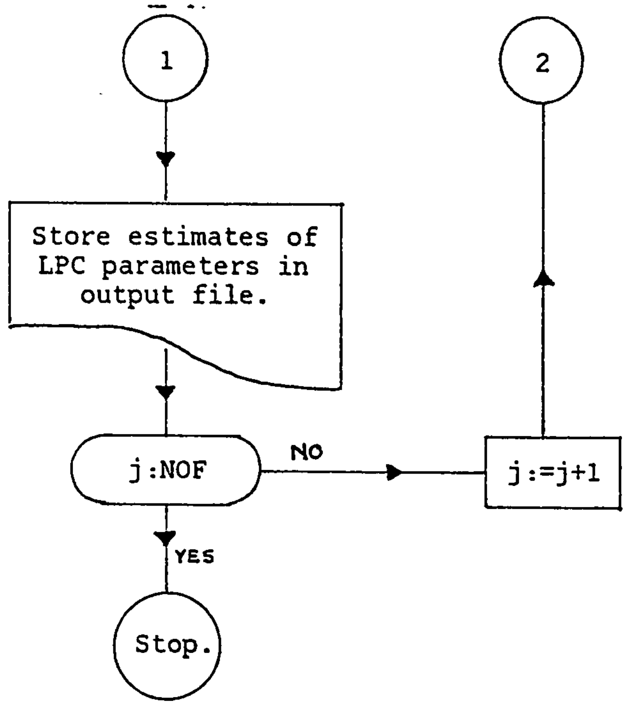
4. Flow Chart for LPC Estimation Using IV Procedure



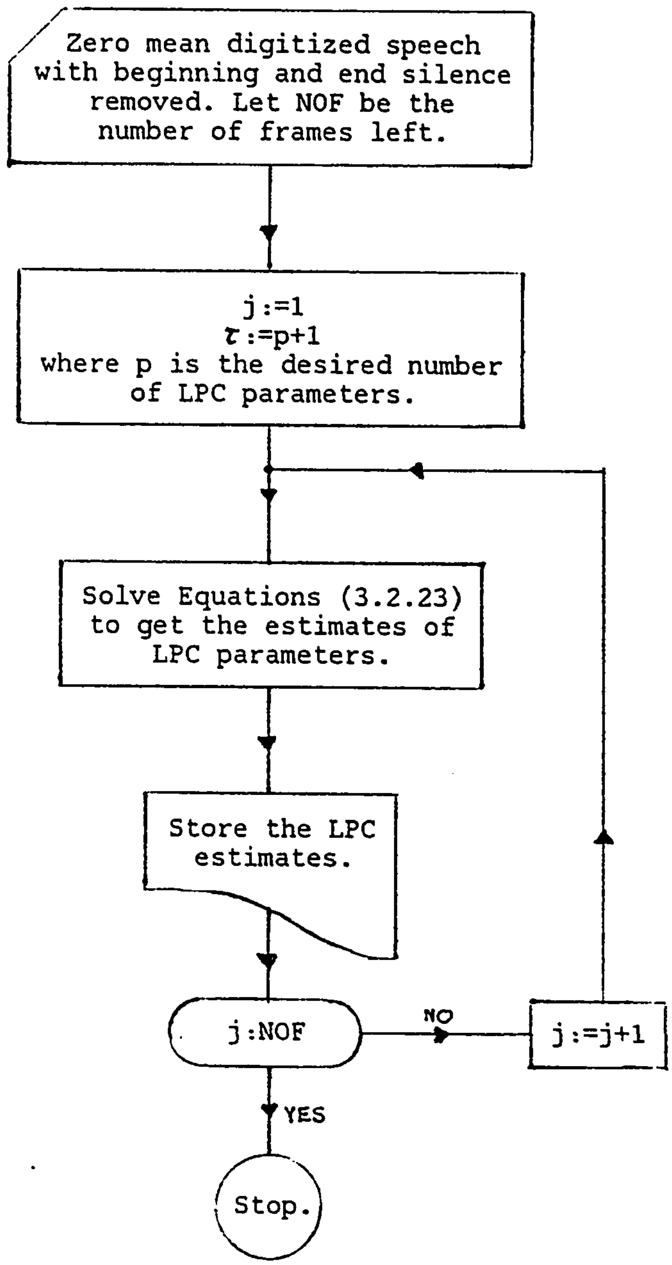

1
2
3
Store the LPC parameters.
j:=j+1
≠
j:NOF
=
Stop.

5. *Flow Chart for LPC Estimation Using AS Method*

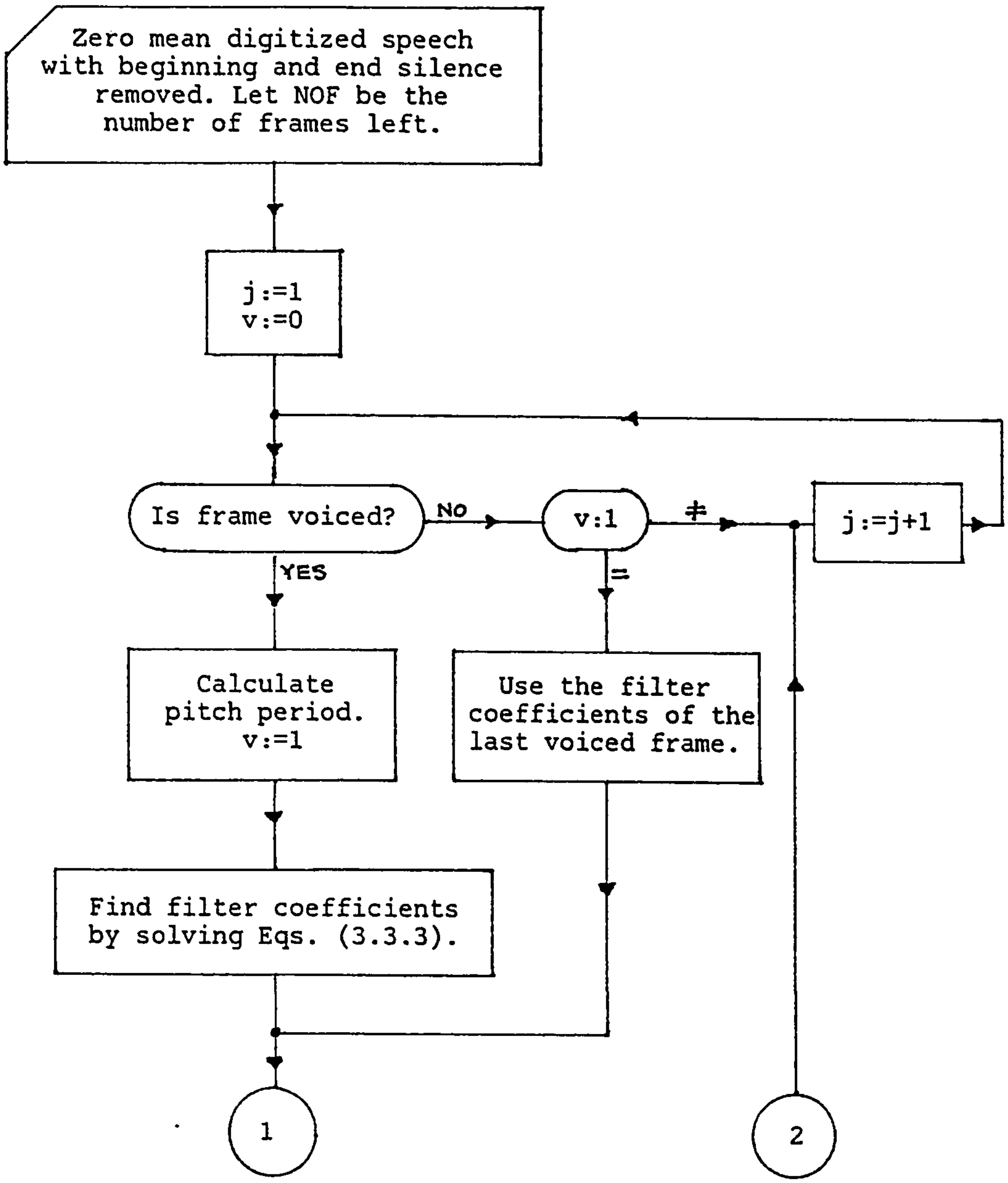


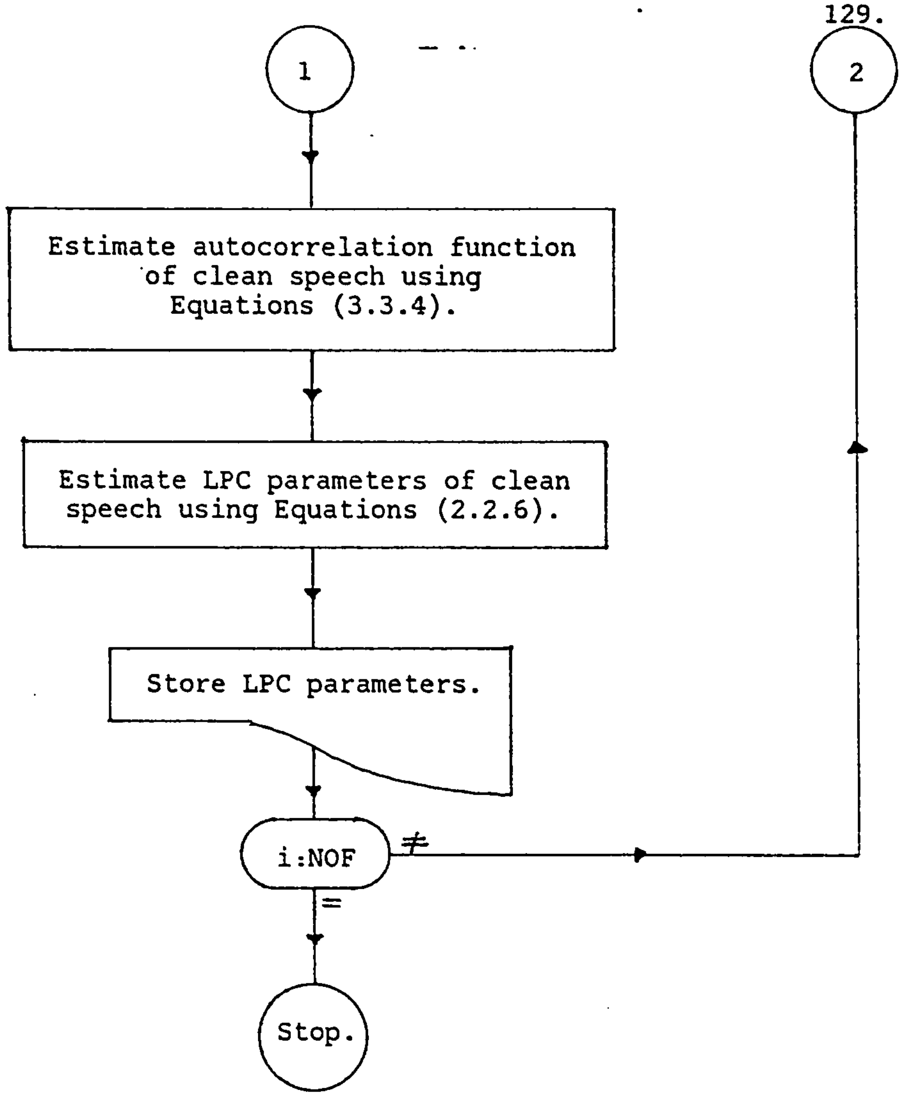

1
2
Store estimates of LPC parameters in output file.
j:NOF
NO
j:=j+1
YES
Stop.

6. Flow Chart for LPC Estimation Using SYW Equations

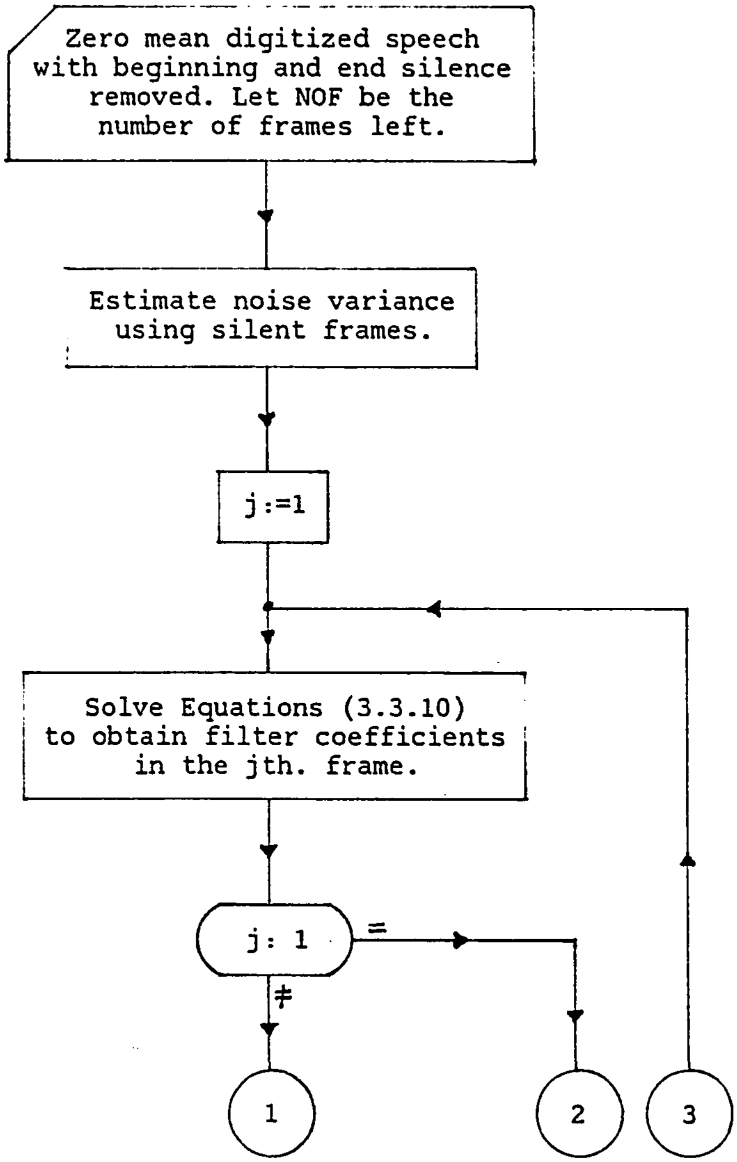


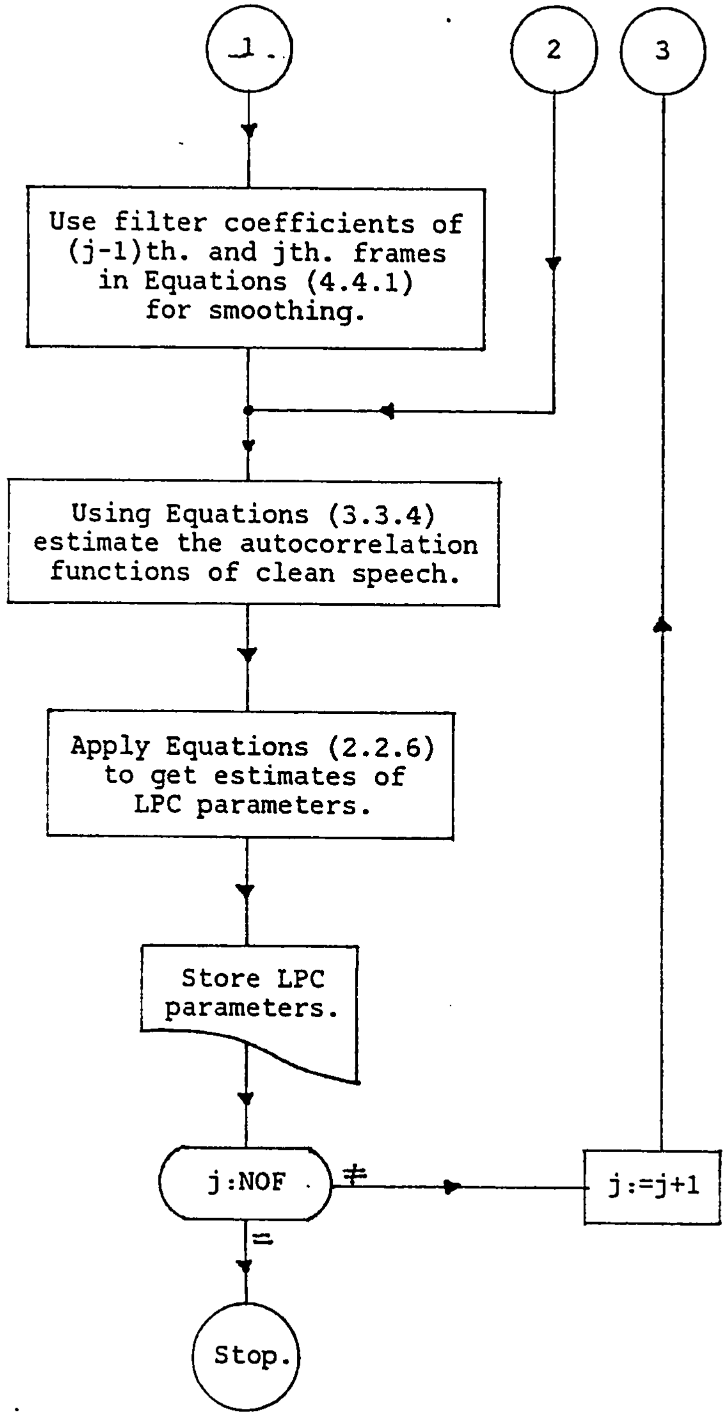
7. *Flow Chart for LPC Estimation Using ANC Procedure*



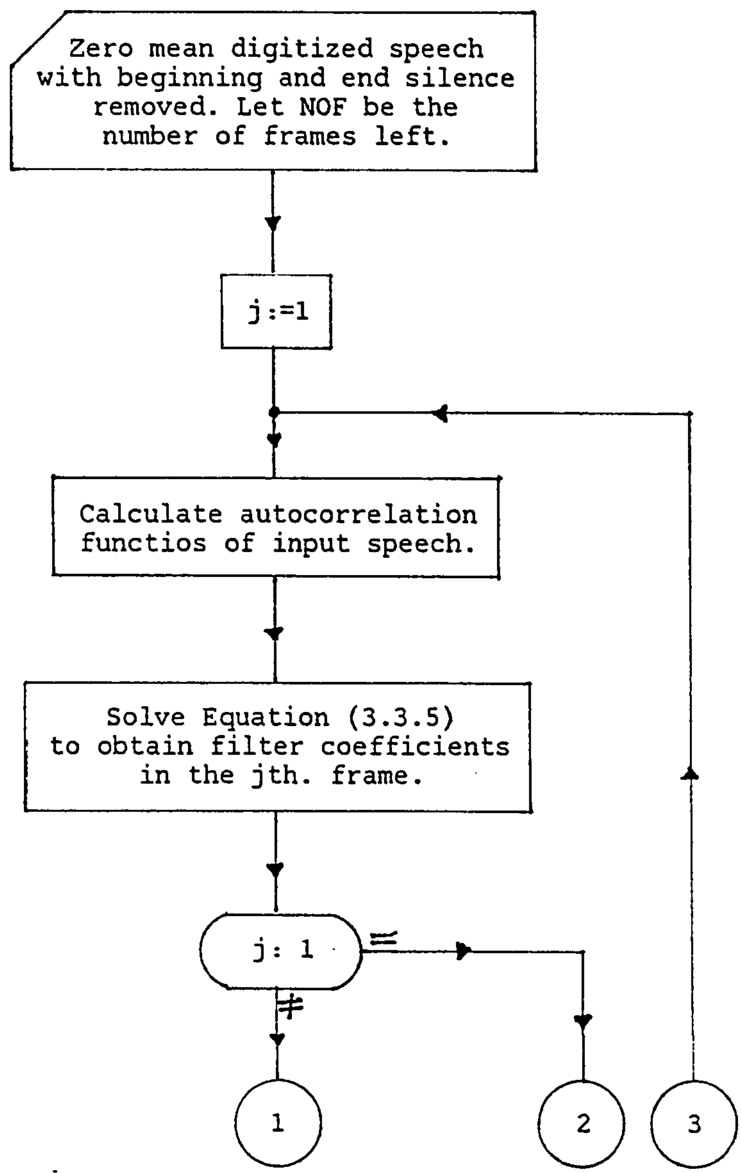

1
2
Estimate autocorrelation function of clean speech using Equations (3.3.4).
Estimate LPC parameters of clean speech using Equations (2.2.6).
Store LPC parameters.
i:NOF
≠
=
Stop.

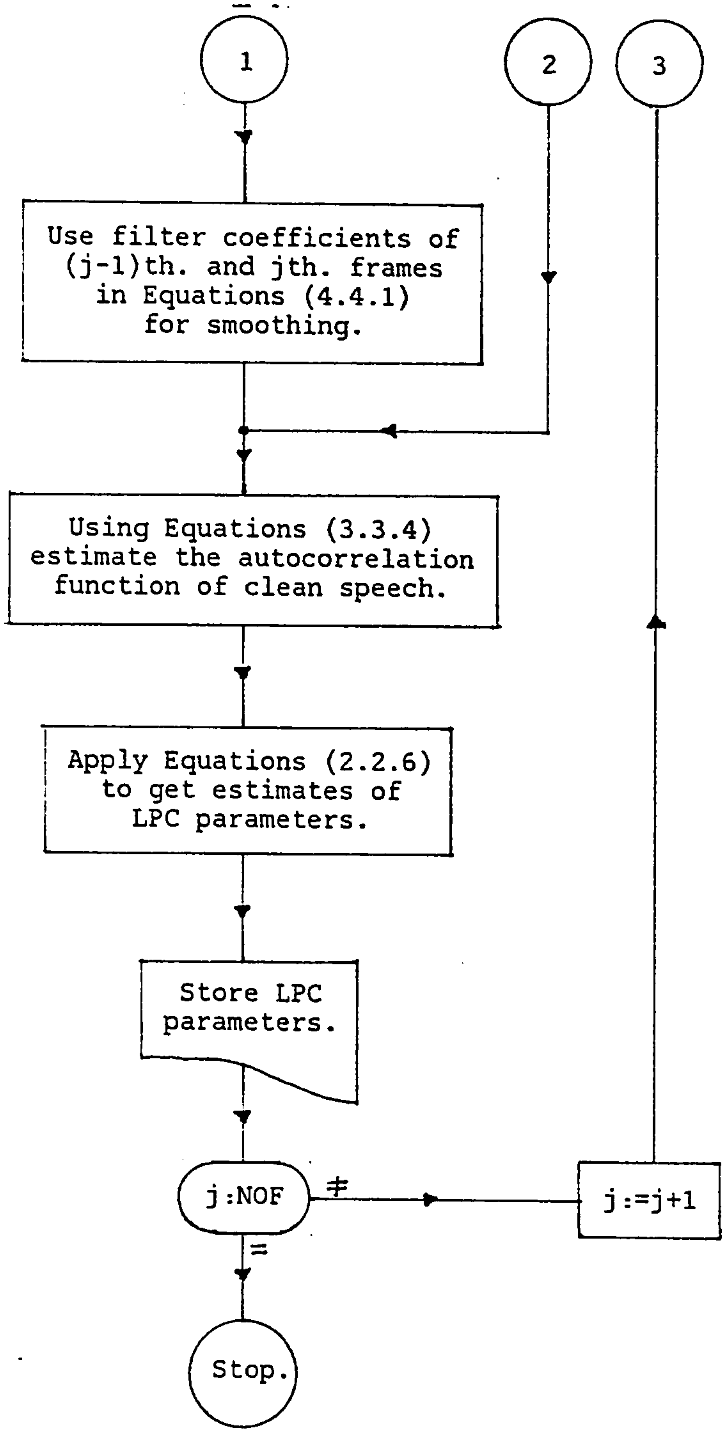
8. *Flow Chart for LPC Estimation Using AFT*




1
2
3
Use filter coefficients of (j-1)th. and jth. frames in Equations (4.4.1) for smoothing.
Using Equations (3.3.4) estimate the autocorrelation functions of clean speech.
Apply Equations (2.2.6) to get estimates of LPC parameters.
Store LPC parameters.
j:NOF
≠
j:=j+1
=
Stop.

9. Flow Chart for LPC Estimation Using LPS




1
2
3
Use filter coefficients of (j-1)th. and jth. frames in Equations (4.4.1) for smoothing.
Using Equations (3.3.4) estimate the autocorrelation function of clean speech.
Apply Equations (2.2.6) to get estimates of LPC parameters.
Store LPC parameters.
j:NOF
≠
j:=j+1
=
Stop.

Appendix H

PROGRAM LISTINGS

1. Introduction

This appendix contains computer programs for the algorithms discussed in the thesis. These programs have been written in WATFIV-S on IBM3033 computer.

The following parameters are used in one or more of these programs:

FLEN: Frame length in number of samples.

FRB: Fraction of the byproduct parameters to be included in the estimates of the LPC parameters (see Appendix B).

FRCVUV & TVUV: Fraction and threshold respectively. Used in v/uv decision. See Appendix D for details.

MAX: Greater than or equal to the number of raw speech samples. Dimensions of the arrays 'X' and 'DATA' should be equal to MAX.

NOP: Number of leading OLPC parameters not to be used in distance calculation for recognition, see page number 38 (NOP=q-1).

NTEST: Number of speakers in the customer (or reference) set, among whom the unknown speaker is to be recognized.

NP: Total number of LPC parameters to be estimated.

NTS: Desired percentage of Gaussian distributed, zero mean, white noise in the signal.

NP: Number of utterances used for creating the reference templates.

PRVFR: Fraction of the enhancement filter coefficients of the previous frame to be combined with those of the current frame for smoothing. See Section 4.4 for details.

SEED: Required by the white noise generating routine 'GAUSS'. Must contain an odd integer with nine or less digits.

TENRGY: Minimum energy of a non-silent frame required to eliminate silent frames from beginning and end of raw speech.

TVUV: See FRCVUV.

2. LPC ESTIMATION USING THE CONVENTIONAL APPROACH (BASED ON THE METHOD OF LEAST SQUARES)

```
C*****************************************************
C=> THIS PROGRAM READS HEXADECIMAL SPEECH SAMPLES
C   FROM DEVICE#10, ESTIMATES LPC PARAMETERS USING
C   THE CONVENTIONAL (LEAST SQUARES) APPROACH, AND
C   STORES THEM ON DEVICE#9. SEE STATEMENT 100 OF
C   SUBROUTINE 'STORE' FOR INPUT FORMAT AND 1200 AND
C   200 OF SUBROUTINE 'LSLPC' FOR OUTPUT FORMAT.
C*****************************************************
      REAL DATA(40000),X(40000),NTS
      INTEGER FLEN,DATA1(40000),SEED
      SEED=65539
      NTS=10.
      TENRGY=12.0
      NP=12
      MAX=40000
      FLEN=200
      CALL STORE(MAX,DATA,X,FLEN,TENRGY,NP,DATA1,SEED,
     &NTS)
      STOP
      END
C*****************************************************
      SUBROUTINE STORE(MAX,DATA,X,FLEN,TENRGY,NP,
     *DATA1,SEED,NTS)
C-----------------------------------------------------
C=> THIS SUBROUTINE READS IN THE RAW DATA, BRINGS
C     THEM TO 0 CENTER, (ASSUMING THAT THE RAW
C     DATA IS IN THE RANGE 0-FF HEX,) DETERMINES FIRST
C     AND LAST SPEECH FRAMES BASED ON THE THRESHOLD
C     'TENRGY'. IT, THEN, CALLS ROUTINE 'CHOP' THAT
C     ELIMINATES THE START AND END SILENT FRAMES.
C-----------------------------------------------------
      REAL DATA(MAX),X(MAX),NTS
      INTEGER FLEN,FRAME,FIRST,DATA1(MAX),SEED
C=> READ IN DATA SAMPLES AND SHIFT THEM TO 0 CENTER.
      READ(10,100,END=1)(DATA1(N),N=1,MAX)
  100 FORMAT(34Z2).
    1 N=N-1
      DO 9 I=1,N
    9 DATA(I)=DATA1(I)
      DO 2 I=1,N
    2  DATA(I)=DATA(I)-128.
      NF=N/FLEN
C=> DETERMINE FIRST AND LAST VOICED FRAME.
```

```
      DO 3 I=1,NF
         FRAME=I
         SS=0.
         DO 4 J=1,FLEN
    4     SS=SS+DATA(FLEN*(FRAME-1)+J)**2
         ENRGY=SS/FLEN
         IF (ENRGY.GE.TENRGY) GOTO 5
    3   CONTINUE
      PRINT,'NO VOICED FRAME FOUND W.R.T. THRESHOLD
     & ENERGY=',TENRGY
      RETURN
    5 FIRST=FRAME
C=> FIRST IS THE NUMBER OF THE FIRST NON-SILENT
C     FRAME.
      DO 6 I=1,NF
         FRAME=NF-I+1
         SS=0.
         DO 7 J=1,FLEN
    7     SS=SS+DATA(FLEN*(FRAME-1)+J)**2
         ENRGY=SS/FLEN
         IF (ENRGY.GE.TENRGY) GOTO 8
    6 CONTINUE
    8 LAST=FRAME
C=> LAST IS THE NUMBER OF THE LAST NON-SILENT
C     FRAME.
      NOF=LAST-FIRST+1
      NOS=NOF*FLEN
      CALL CHOP(DATA,MAX,X,NOS,FIRST,FLEN,NOF,NP,SEED,
     &NTS)
      RETURN
      END
C*******************************************************
      SUBROUTINE CHOP(DATA,MAX,X,NOS,FIRST,FLEN,
     *NOF,NP,SEED,NTS)
C-------------------------------------------------------
C=> THIS SUBROUTINE ELIMINATES START AND END
C     SILENT FRAMES AND STORES THE REMAINING
C     SAMPLES IN ARRAY 'X'.
C-------------------------------------------------------
      REAL DATA(MAX),X(NOS),R(20),RHO(20),PHI(20,20),NTS
      INTEGER FIRST,FLEN,SEED
      DO 9 I=1,NOS
    9  X(I)=DATA(FLEN*(FIRST-1)+I)
      NPP=NP+1
      CALL LSLPC(NOF,NOS,X,FLEN,NP,NPP,R,RHO,PHI,SEED,
     &NTS)
      RETURN
      END
```

```
C*******************************************************
      SUBROUTINE LSLPC(NOF,NOS,X,FLEN,NP,NPP,R,RHO,PHI,
     &SEED,NTS)
C-------------------------------------------------------
C=> THIS SUBROUTINE ADDS NOISE TO SPEECH USING
C     SUBROUTINE 'GAUSS' AND ESTIMATES LPC PARAMETERS
C     USING METHOD OF LEAST SQUARES. THE LPC
C     PARAMETERS ARE STORED ON DEVICE#9. SEE
C     STATEMENTS 1200 AND 200 FOR OUTPUT FORMAT.
C-------------------------------------------------------
      REAL R(NPP),RHO(NP),PHI(NP,NP),X(NOS),NTS,NPHI,
     &MEAN
      INTEGER FRAME,FLEN,SEED
C=> WRITE THE TOTAL NUMBER OF FRAMES.
      WRITE(9,300)NOF
  300 FORMAT(I3)
C=> FIND MEAN & SD. OF THE SPEECH DATA.
      SUM=SUMS=0.
      DO 19 I=1,NOS
        SUM=SUM+X(I)
   19   SUMS=SUMS+X(I)**2
      MEAN=SUM/NOS
      SD=SQRT((SUMS/NOS)-(MEAN**2))
C=> S IS THE STANDARD DEVIATION OF NOISE TO BE
C     ADDED TO SPEECH.
      S=NTS*SD/100.
C=> ADD NOISE TO SPEECH.
      DO 20 I=1,NOS
        CALL GAUSS(SEED,S,0.0,V)
   20   X(I)=X(I)+V
      DO 10 FRAME=1,NOF
        DO 11 I=1,NPP
   11     R(I)=0.
        DO 12 I=1,NP
          RHO(I)=0.
          DO 12 J=1,NP
   12       PHI(I,J)=0.
C=> CALCULATE THE AUTOCORRELATION FUNCTIONS.
        DO 14 K=1,NPP
          KFLEN=FLEN-K+1
          DO 15 I=1,KFLEN
   15       R(K)=R(K)+X(FLEN*(FRAME-1)+I)*
     &            X(FLEN*(FRAME-1)+I+K-1)
   14     R(K)=R(K)/FLEN
        DO 16 I=1,NP
   16     RHO(I)=R(I+1)/R(1)
C=> CALCULATE THE LPC PARAMETERS AND STORE
C     ON DEVICE#9.
```

```
      PHI(1,1)=RHO(1)
      DO 17 I=2,NP
        NPHI=RHO(I)
        DPHI=1.
        IM1=I-1
        DO 18 J=1,IM1
          NPHI=NPHI-PHI(I-1,J)*RHO(I-J)
   18     DPHI=DPHI-PHI(I-1,J)*RHO(J)
        PHI(I,I)=NPHI/DPHI
        DO 17 J=1,IM1
   17     PHI(I,J)=PHI(I-1,J)-PHI(I,I)*PHI(I-1,I-J)
C WRITING OUTPUT IN PROPER FILES
   10   WRITE(9,200)(PHI(NP,I),I=1,NP)
  200 FORMAT(6E15.7/6E15.7)
      RETURN
      END
C*******************************************************
      SUBROUTINE GAUSS(IX,S,AM,V)
C-------------------------------------------------------
C=> THIS ROUTINE PRODUCES GAUSSIAN DISTRIBUTED NOISE WITH
C     THE HELP OF THE ROUTINE 'RANDU'.
C-------------------------------------------------------
      A=0.0
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
   50 A=A+Y
      V=(A-6.0)*S+AM
      RETURN
      END
C*******************************************************
      SUBROUTINE RANDU(IX,IY,YFL)
      IY=IX*16807
      IF(IY)5,6,6
    5 IY=IY+2147483647
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
C*******************************************************
```

3. LPC ESTIMATION USING IV

```
C*****************************************************
C=> THIS PROGRAM READS HEXADECIMAL SPEECH SAMPLES
C   FROM DEVICE#10, ESTIMATES LPC PARAMETERS USING
C   THE INSTRUMENTAL VARIABLES (IV) METHOD, AND
C   STORES THEM ON DEVICE#9. SEE STATEMENT 100 OF
C   SUBROUTINE 'STORE' FOR INPUT FORMAT AND 1200 AND
C   200 OF SUBROUTINE 'IVLPC' FOR OUTPUT FORMAT.
C*****************************************************
      REAL DATA(40000),X(40000),NTS
      INTEGER FLEN,DATA1(40000),SEED
      COMMON FRCVUV,TVUV,FRB
      SEED=65539
      FRCVUV=3.
      TVUV=2.
      NTS=0.
      TENRGY=9.0
      FRB=.5
      NP=12
      MAX=40000
      FLEN=200
      CALL STORE(MAX,DATA,X,FLEN,TENRGY,NP,DATA1,SEED,
     &NTS)
      STOP
      END
C*****************************************************
      SUBROUTINE STORE(MAX,DATA,X,FLEN,TENRGY,NP,
     *DATA1,SEED,NTS)
C-----------------------------------------------------
C=> THIS SUBROUTINE READS IN THE RAW DATA, BRINGS
C     THEM TO 0 CENTER, (ASSUMING THAT THE RAW
C     DATA IS IN THE RANGE 0-FF HEX,) DETERMINES FIRST
C     AND LAST SPEECH FRAMES BASED ON THE THRESHOLD
C     'TENRGY'. IT, THEN, CALLS ROUTINE 'CHOP' THAT
C     ELIMINATES THE START AND END SILENT FRAMES.
C-----------------------------------------------------
      REAL DATA(MAX),X(MAX),NTS
      INTEGER FLEN,FRAME,FIRST,DATA1(MAX),SEED
C=> READ IN DATA SAMPLES AND SHIFT THEM TO 0 CENTER.
      READ(10,100,END=1)(DATA1(N),N=1,MAX)
  100 FORMAT(34Z2)
    1 N=N-1
      DO 9 I=1,N
    9 DATA(I)=DATA1(I)
      DO 2 I=1,N
    2  DATA(I)=DATA(I)-128.
```

```
      NF=N/FLEN
C=> DETERMINE FIRST AND LAST VOICED FRAME.
      DO 3 I=1,NF
        FRAME=I
        SS=0.
        DO 4 J=1,FLEN
    4    SS=SS+DATA(FLEN*(FRAME-1)+J)**2
        ENRGY=SS/FLEN
        IF (ENRGY.GE.TENRGY) GOTO 5
    3   CONTINUE
      PRINT,'NO VOICED FRAME FOUND W.R.T. THRESHOLD
     & ENERGY=',TENRGY
      RETURN
    5 FIRST=FRAME
C=> FIRST IS THE NUMBER OF THE FIRST NON-SILENT
C     FRAME.
      DO 6 I=1,NF
        FRAME=NF-I+1
        SS=0.
        DO 7 J=1,FLEN
    7    SS=SS+DATA(FLEN*(FRAME-1)+J)**2
        ENRGY=SS/FLEN
        IF (ENRGY.GE.TENRGY) GOTO 8
    6 CONTINUE
    8 LAST=FRAME
C=> LAST IS THE NUMBER OF THE LAST NON-SILENT
C     FRAME.
      NOF=LAST-FIRST+1
      NOS=NOF*FLEN
      CALL CHOP(DATA,MAX,X,NOS,FIRST,FLEN,NOF,NP,SEED,
     &NTS)
      RETURN
      END
C*****************************************************
      SUBROUTINE CHOP(DATA,MAX,X,NOS,FIRST,FLEN,
     *NOF,NP,SEED,NTS)
C----------------------------------------------------
C=> THIS SUBROUTINE ELIMINATES START AND END
C     SILENT FRAMES AND STORES THE REMAINING
C     SAMPLES IN ARRAY 'X'.
C----------------------------------------------------
      REAL DATA(MAX),X(NOS),PHI(20,20),NTS,COMB(20)
      INTEGER FIRST,FLEN,SEED
      DO 9 I=1,NOS
    9  X(I)=DATA(FLEN*(FIRST-1)+I)
      CALL IVLPC(NOF,NOS,X,FLEN,NP,PHI,SEED,NTS,COMB)
      RETURN
      END
```

```
C*****************************************************
      SUBROUTINE IVLPC(NOF,NOS,X,FLEN,NP,PHI,SEED,
     *NTS,COMB)
C-----------------------------------------------------
C=> THIS SUBROUTINE ADDS NOISE TO SPEECH USING
C     SUBROUTINE 'GAUSS', ENHANCES SPEECH USING
C     'AFT' AND CALCULATES LPC PARAMETERS. THE LPC
C     PARAMETERS ARE STCRED ON DEVICE#9. SEE
C     STATEMENTS 1200 AND 200 FOR OUTPUT FORMAT.
C-----------------------------------------------------
      COMMON FRCVUV,TVUV,FRB
      REAL RHO(212),PHI(NP,NP),X(NOS),NTS,NPHI,MEAN,
     &PHIR(200),NB,B(12,12),COMB(NP)
      INTEGER FRAME,FLEN,SEED,PLEN,FLENMK,PEAK,PMNP,
     &PPNP
C=> WRITE THE TOTAL NUMBER OF FRAMES.
      WRITE(9,1200)NOF
 1200 FORMAT(I3)
C=> FIND MEAN & SD. OF THE SPEECH DATA.
      IF (NTS.LT.1.E-4) GOTO 30
      SUM=SUMS=0.
      DO 19 I=1,NOS
        SUM=SUM+X(I)
   19   SUMS=SUMS+X(I)**2
      MEAN=SUM/NOS
      SD=SQRT((SUMS/NOS)-(MEAN**2))
C=> S IS THE STANDARD DEVIATION OF NOISE TO BE
C     ADDED TO SPEECH.
      S=NTS*SD/100.
C=> ADD NOISE TO SPEECH.
      DO 20 I=1,NOS
        CALL GAUSS(SEED,S,0.0,V)
   20   X(I)=X(I)+V
   30 DO 21 FRAME=1,NOF
        MAXP=0
        PEAK=1
C=> CALCULATE THE AUTOCORRELATION FUNCTION.
        SUMS=0.
        DO 22 I=1,FLEN
   22     SUMS=SUMS+X(FLEN*(FRAME-1)+I)**2
        ENRGY=SUMS/FLEN
C ENRGY IS ALSO = PHIR(0)
   33   DO 26 K=1,175
          FLENMK=FLEN-K
          PHIR(K)=0.
          DO 23 J=1,FLENMK
   23      PHIR(K)=PHIR(K)+X(FLEN*(FRAME-1)+J)*
     &       X(FLEN*(FRAME-1)+J+K)
```

```
   26      PHIR(K)=PHIR(K)/FLEN
C=> CHECK FOR VOICED OR NOT AND ESTIMATE PITCH IF
C      FRAME IS VOICED.
           GPMAX=0.
           DO 24 PLEN=24,120
              N=FLEN/PLEN
              NM1=N-1
              IF(NM1.EQ.0)NM1=1
              GP=0.
              DO 25 L=1,NM1
   25          GP=GP+PHIR(L*PLEN)
              GP=2.*PLEN*GP/FLEN
              IF(GP.GT.GPMAX)THEN
                 GPMAX=GP
                 MAXP=PLEN
                 ENDIF
   24    CONTINUE
          IF (GPMAX.LE.0.) GOTO 21
          VUV=FRCVUV*PHIR(MAXP)/ENRGY+GPMAX/ENRGY
          IF(VUV.LT.TVUV) GOTO 21
C=> CALCULATE LPC'S FROM PEAK#2 & THE BYPRODUCT
C      PARAMETERS, ADD WITH APPROPRIATE WEIGHTS
C      I.E. 1.-FRB AND FRB RESPECTIVELY AND STORE
C      IN COMB(I).
   48      PEAK=2
        PMNP=MAXP-NP
        PPNP=MAXP+NP
        DO 36 I=PMNP,PPNP
           IF (MAXP.GT.I) THEN
              PHIR(I)=PHIR(I)*FLEN*(FLEN-2.*MAXP+I)/
     &       ((FLEN-MAXP)*(FLEN-I))
              ELSE
              PHIR(I)=PHIR(I)*FLEN/(FLEN-MAXP)
              ENDIF
   36 CONTINUE
        DO 43 I=PMNP,PPNP
   43    RHO(I)=PHIR(I)/PHIR(MAXP)
        DO 37 I=1,12
           DO 37 J=1,12
              PHI(I,J)=0.
   37       B(I,J)=0.
        PHI(1,1)=RHO(MAXP+1);B(1,1)=RHO(MAXP-1)
        K=1
   42 NPHI=RHO(MAXP+K+1)
        DPHI=1.
        DO 38 J=1,K
           NPHI=NPHI-RHO(MAXP+J)*PHI(K,K+1-J)
   38    DPHI=DPHI-RHO(MAXP+J)*B(K,J)
```

```
      PHI(K+1,K+1)=NPHI/DPHI
      DO 39 I=1,K
   39   PHI(K+1,I)=PHI(K,I)-B(K,K+1-I)*PHI(K+1,K+1)
      NB=RHO(MAXP-K-1)
      DB=1.
      DO 40 J=1,K
         NB=NB-RHO(MAXP-J)*B(K,K+1-J)
   40   DB=DB-RHO(MAXP-J)*PHI(K,J)
      B(K+1,K+1)=NB/DB
      DO 41 I=1,K
   41 B(K+1,I)=B(K,I)-PHI(K,K+1-I)*B(K+1,K+1)
      IF ((K+1).EQ.NP) GOTO 10
      K=K+1
      GOTO 42
   10 DO 47 I=1,NP
   47   COMB(I)=(1.-FRB)*PHI(NP,I)+FRB*B(NP,I)
         WRITE(9,200)(COMB(I),I=1,NP)
  200 FORMAT(6E15.7/6E15.7)
   21 CONTINUE
      RETURN
      END
C*************************************************************
      SUBROUTINE GAUSS(IX,S,AM,V)
C-------------------------------------------------------------
C=> THIS ROUTINE PRODUCES GAUSSIAN DISTRIBUTED NOISE WITH
C    THE HELP OF THE ROUTINE 'RANDU'.
C-------------------------------------------------------------
      A=0.0
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
   50 A=A+Y
      V=(A-6.0)*S+AM
      RETURN
      END
C*************************************************************
      SUBROUTINE RANDU(IX,IY,YFL)
      IY=IX*16807
      IF(IY)5,6,6
    5 IY=IY+2147483647
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
C*************************************************************
```

4. LPC ESTIMATION USING AS

```
C*****************************************************
C=> THIS PROGRAM READS HEXADECIMAL SPEECH SAMPLES
C   FROM DEVICE#10, ESTIMATES LPC PARAMETERS USING
C   THE AUTOCORRELATION SUBTRACTION (AS) METHOD, AND
C   STORES THEM ON DEVICE#9. SEE STATEMENT 100 OF
C   SUBROUTINE 'STORE' FOR INPUT FORMAT AND 1200 AND
C   200 OF SUBROUTINE 'ASLPC' FOR OUTPUT FORMAT.
C*****************************************************
      REAL DATA(40000),X(40000),NTS
      INTEGER FLEN,DATA1(40000),SEED
      SEED=65539
      NTS=10.
      TENRGY=12.0
      NP=12
      MAX=40000
      FLEN=200
      CALL STORE(MAX,DATA,X,FLEN,TENRGY,NP,DATA1,SEED,
     &NTS)
      STOP
      END
C*****************************************************
      SUBROUTINE STORE(MAX,DATA,X,FLEN,TENRGY,NP,
     *DATA1,SEED,NTS)
C-----------------------------------------------------
C=> THIS SUBROUTINE READS IN THE RAW DATA, BRINGS
C     THEM TO 0 CENTER, (ASSUMING THAT THE RAW
C     DATA IS IN THE RANGE 0-FF HEX,) DETERMINES FIRST
C     AND LAST SPEECH FRAMES BASED ON THE THRESHOLD
C     'TENRGY'. IT, THEN, CALLS ROUTINE 'CHOP' THAT
C     ELIMINATES THE START AND END SILENT FRAMES.
C-----------------------------------------------------
      REAL DATA(MAX),X(MAX),NTS
      INTEGER FLEN,FRAME,FIRST,DATA1(MAX),SEED
C=> READ IN DATA SAMPLES AND SHIFT THEM TO 0 CENTER.
      READ(10,100,END=1)(DATA1(N),N=1,MAX)
  100 FORMAT(34Z2)
    1 N=N-1
      DO 9 I=1,N
    9 DATA(I)=DATA1(I)
      DO 2 I=1,N
    2  DATA(I)=DATA(I)-128.
      NF=N/FLEN
C=> DETERMINE FIRST AND LAST VOICED FRAME.
      DO 3 I=1,NF
        FRAME=I
```

```
        SS=0.
        DO 4 J=1,FLEN
    4     SS=SS+DATA(FLEN*(FRAME-1)+J)**2
        ENRGY=SS/FLEN
        IF (ENRGY.GE.TENRGY) GOTO 5
    3   CONTINUE
      PRINT,'NO VOICED FRAME FOUND W.R.T. THRESHOLD
     & ENERGY',TENRGY
      RETURN
    5 FIRST=FRAME
C=> FIRST IS THE NUMBER OF THE FIRST NON-SILENT
C     FRAME.
      DO 6 I=1,NF
        FRAME=NF-I+1
        SS=0.
        DO 7 J=1,FLEN
    7     SS=SS+DATA(FLEN*(FRAME-1)+J)**2
        ENRGY=SS/FLEN
        IF (ENRGY.GE.TENRGY) GOTO 8
    6 CONTINUE
    8 LAST=FRAME
C=> LAST IS THE NUMBER OF THE LAST NON-SILENT
C     FRAME.
      NOF=LAST-FIRST+1
      NOS=NOF*FLEN
      CALL CHOP(DATA,MAX,X,NOS,FIRST,FLEN,NOF,NP,SEED,
     &NTS)
      RETURN
      END
C**********************************************************
      SUBROUTINE CHOP(DATA,MAX,X,NOS,FIRST,FLEN,
     *NOF,NP,SEED,NTS)
C---------------------------------------------------------
C=> THIS SUBROUTINE ELIMINATES START AND END
C     SILENT FRAMES AND STORES THE REMAINING
C     SAMPLES IN ARRAY 'X'.
C---------------------------------------------------------
      REAL DATA(MAX),X(NOS),R(20),RHO(20),PHI(20,20),NTS
      INTEGER FIRST,FLEN,SEED
      DO 9 I=1,NOS
    9   X(I)=DATA(FLEN*(FIRST-1)+I)
      NPP=NP+1
      CALL ASLPC(NOF,NOS,X,FLEN,NP,NPP,R,RHO,PHI,SEED,
     &NTS)
      RETURN
      END
```

```
C*****************************************************
      SUBROUTINE ASLPC(NOF,NOS,X,FLEN,NP,NPP,PHI,SEED,
     *NTS)
C-----------------------------------------------------
C=> THIS SUBROUTINE ADDS NOISE TO SPEECH USING
C     SUBROUTINE 'GAUSS', ENHANCES SPEECH USING
C     'AFT' AND CALCULATES LPC PARAMETERS. THE LPC
C     PARAMETERS ARE STORED ON DEVICE#9. SEE
C     STATEMENTS 1200 AND 200 FOR OUTPUT FORMAT.
C-----------------------------------------------------
      REAL R(NPP),RHO(NP),PHI(NP,NP),X(NOS),NTS,NPHI,
     &MEAN
      INTEGER FRAME,FLEN,SEED
C=> WRITE THE TOTAL NUMBER OF FRAMES.
      WRITE(9,300)NOF
  300 FORMAT(I3)
C=> FIND MEAN & SD. OF THE SPEECH DATA.
      S=0.
      IF (NTS.EQ.0.) GOTO 40
      SUM=SUMS=0.
      DO 19 I=1,NOS
        SUM=SUM+X(I)
   19   SUMS=SUMS+X(I)**2
      MEAN=SUM/NOS
      SD=SQRT((SUMS/NOS)-(MEAN**2))
C=> S IS THE STANDARD DEVIATION OF NOISE TO BE
C     ADDED TO SPEECH.
      S=NTS*SD/100.
C=> ADD NOISE TO SPEECH.
      DO 20 I=1,NOS
        CALL GAUSS(SEED,S,0.0,V)
   20   X(I)=X(I)+V
   40 DO 10 FRAME=1,NOF
C INITIALIZATION
        DO 11 I=1,NPP
   11     R(I)=0.
        DO 12 I=1,NP
          RHO(I)=0.
          DO 12 J=1,NP
   12       PHI(I,J)=0.
C=> CALCULATE THE AUTOCORRELATION FUNCTION.
        DO 14 K=1,NPP
          KFLEN=FLEN-K+1
          DO 15 I=1,KFLEN
   15       R(K)=R(K)+X(FLEN*(FRAME-1)+I)*
     &              X(FLEN*(FRAME-1)+I+K-1)
   14     R(K)=R(K)/FLEN
        R(1)=R(1)-S**2
```

```
      DO 16 I=1,NP
16     RHO(I)=R(I+1)/R(1)
      PHI(1,1)=RHO(1)
      DO 17 I=2,NP
        NPHI=RHO(I)
        DPHI=1.
        IM1=I-1
        DO 18 J=1,IM1
          NPHI=NPHI-PHI(I-1,J)*RHO(I-J)
18       DPHI=DPHI-PHI(I-1,J)*RHO(J)
        PHI(I,I)=NPHI/DPHI
        DO 17 J=1,IM1
17       PHI(I,J)=PHI(I-1,J)-PHI(I,I)*PHI(I-1,I-J)
10    WRITE(9,200)(PHI(NP,I),I=1,NP)
200 FORMAT(6E15.7/6E15.7)
      RETURN
      END
C*************************************************************
      SUBROUTINE GAUSS(IX,S,AM,V)
C-------------------------------------------------------------
C=> THIS ROUTINE PRODUCES GAUSSIAN DISTRIBUTED NOISE WITH
C     THE HELP OF THE ROUTINE 'RANDU'.
C-------------------------------------------------------------
      A=0.0
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
50  A=A+Y
      V=(A-6.0)*S+AM
      RETURN
      END
C*************************************************************
      SUBROUTINE RANDU(IX,IY,YFL)
      IY=IX*16807
      IF(IY)5,6,6
    5 IY=IY+2147483647
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
C*************************************************************
```

5. LPC ESTIMATION USING SYW EQUATIONS

```
C*******************************************************
C=> THIS PROGRAM READS HEXADECIMAL SPEECH SAMPLES
C   FROM DEVICE#10, ESTIMATES LPC PARAMETERS USING
C   THE SHIFTED YULE-WALKER (SYW) EQUATIONS, AND
C   STORES THEM ON DEVICE#9. SEE STATEMENT 100 OF
C   SUBROUTINE 'STORE' FOR INPUT FORMAT AND 1200 AND
C   200 OF SUBROUTINE 'SYWLPC' FOR OUTPUT FORMAT.
C*******************************************************
      REAL DATA(40000),X(40000),NTS
      INTEGER FLEN,DATA1(40000),SEED
      SEED=65539
      NTS=0.
      TENRGY=9.0
      NP=12
      MAX=40000
      FLEN=200
      CALL STORE(MAX,DATA,X,FLEN,TENRGY,NP,DATA1,SEED,
     &NTS)
      STOP
      END
C*******************************************************
      SUBROUTINE STORE(MAX,DATA,X,FLEN,TENRGY,NP,
     *DATA1,SEED,NTS)
C-------------------------------------------------------
C=> THIS SUBROUTINE READS IN THE RAW DATA, BRINGS
C     THEM TO 0 CENTER, (ASSUMING THAT THE RAW
C     DATA IS IN THE RANGE 0-FF HEX,) DETERMINES FIRST
C     AND LAST SPEECH FRAMES BASED ON THE THRESHOLD
C     'TENRGY'. IT, THEN, CALLS ROUTINE 'CHOP' THAT
C     ELIMINATES THE START AND END SILENT FRAMES.
C-------------------------------------------------------
      REAL DATA(MAX),X(MAX),NTS
      INTEGER FLEN,FRAME,FIRST,DATA1(MAX),SEED
C=> READ IN DATA SAMPLES AND SHIFT THEM TO 0 CENTER.
      READ(10,100,END=1)(DATA1(N),N=1,MAX)
  100 FORMAT(34Z2)
    1 N=N-1
      DO 9 I=1,N
    9 DATA(I)=DATA1(I)
      DO 2 I=1,N
    2  DATA(I)=DATA(I)-128.
      NF=N/FLEN
C=> DETERMINE FIRST AND LAST VOICED FRAME.
      DO 3 I=1,NF
        FRAME=I
```

```
      SS=0.
      DO 4 J=1,FLEN
 4     SS=SS+DATA(FLEN*(FRAME-1)+J)**2
      ENRGY=SS/FLEN
      IF (ENRGY.GE.TENRGY) GOTO 5
 3   CONTINUE
    PRINT,'NO VOICED FRAME FOUND W.R.T. THRESHOLD
   & ENERGY=',TENRGY
    RETURN
 5 FIRST=FRAME
C=> FIRST IS THE NUMBER OF THE FIRST NON-SILENT
C     FRAME.
    DO 6 I=1,NF
      FRAME=NF-I+1
      SS=0.
      DO 7 J=1,FLEN
 7     SS=SS+DATA(FLEN*(FRAME-1)+J)**2
      ENRGY=SS/FLEN
      IF (ENRGY.GE.TENRGY) GOTO 8
 6 CONTINUE
 8 LAST=FRAME
C=> LAST IS THE NUMBER OF THE LAST NON-SILENT
C     FRAME.
    NOF=LAST-FIRST+1
    NOS=NOF*FLEN
    CALL CHOP(DATA,MAX,X,NOS,FIRST,FLEN,NOF,NP,SEED,
   &NTS)
    RETURN
    END
C*******************************************************
    SUBROUTINE CHOP(DATA,MAX,X,NOS,FIRST,FLEN,
   *NOF,NP,SEED,NTS)
C-------------------------------------------------------
C=> THIS SUBROUTINE ELIMINATES START AND END
C     SILENT FRAMES AND STORES THE REMAINING
C     SAMPLES IN ARRAY 'X'.
C-------------------------------------------------------
    REAL DATA(MAX),X(NOS),PHI(20,20),NTS
    INTEGER FIRST,FLEN,SEED
    DO 9 I=1,NOS
 9   X(I)=DATA(FLEN*(FIRST-1)+I)
    CALL SYWLPC(NOF,NOS,X,FLEN,NP,PHI,SEED,NTS)
    RETURN
    END
C*******************************************************
```

```
      SUBROUTINE SYWLPC(NOF,NOS,X,FLEN,NP,PHI,SEED,
     *NTS)
C-----------------------------------------------------
C=> THIS SUBROUTINE ADDS NOISE TO SPEECH USING
C     SUBROUTINE 'GAUSS' AND CALCULATES LPC PARAMETERS
C     USING SHIFTED YULE-WALKER EQUATIONS AND STORES
C     THEM ON DEVICE#9.
C     SEE STATEMENTS 1200 AND 200 FOR OUTPUT FORMAT.
C-----------------------------------------------------
      REAL RHO(25),PHI(NP,NP),X(NOS),NTS,NPHI,MEAN,
     &PHIR(25),NB,NC,B(12,12),C(12,12),H(12)
      INTEGER FRAME,FLEN,SEED,FLENMK
C=> WRITE THE TOTAL NUMBER OF FRAMES.
      WRITE(9,1200)NOF
 1200 FORMAT(I3)
C=> FIND MEAN & SD. OF THE SPEECH DATA.
      S=0.
      IF (NTS.LT.1.) GOTO 30
      SUM=SUMS=0.
      DO 19 I=1,NOS
        SUM=SUM+X(I)
   19   SUMS=SUMS+X(I)**2
      MEAN=SUM/NOS
      SD=SQRT((SUMS/NOS)-(MEAN**2))
      S=NTS*SD/100.
C=> S IS THE STANDARD DEVIATION OF NOISE TO BE
C     ADDED TO SPEECH.
C=> ADD NOISE TO SPEECH.
      DO 20 I=1,NOS
        CALL GAUSS(SEED,S,0.0,V)
   20   X(I)=X(I)+V
   30 NPT2P1=NP*2+1
      DO 21 FRAME=1,NOF
C=> CALCULATE THE AUTOCORRELATION FUNCTION.
   33   DO 26 K=1,NPT2P1
          FLENMK=FLEN-K
          PHIR(K)=0.
          DO 23 J=1,FLENMK
   23       PHIR(K)=PHIR(K)+X(FLEN*(FRAME-1)+J)*
     &        X(FLEN*(FRAME-1)+J+K)
   26     PHIR(K)=PHIR(K)/FLEN
C=> GET B(13,I),I=1,...,13
      NPP1=NP+1
      DO 43 I=1,NPT2P1
   43   RHO(I)=PHIR(I)/PHIR(NP+1)
      DO 37 I=1,NP
        DO 37 J=1,NP
          B(I,J)=0.
```

```
 37      C(I,J)=0.
      B(1,1)=RHO(NP+2);C(1,1)=RHO(NP)
      K=1
 42  NB=RHO(NP+K+2)
      DB=1.
      DO 38 J=1,K
        NB=NB-RHO(NP+1+J)*B(K,K+1-J)
 38    DB=DB-RHO(NP+1+J)*C(K,J)
      B(K+1,K+1)=NB/DB
      DO 39 I=1,K
 39    B(K+1,I)=B(K,I)-C(K,K+1-I)*B(K+1,K+1)
      NC=RHO(NP-K)
      DC=1.
      DO 40 J=1,K
        NC=NC-RHO(NP+1-J)*C(K,K+1-J)
 40    DC=DC-RHO(NP+1-J)*B(K,J)
      C(K+1,K+1)=NC/DC
      DO 41 I=1,K
 41  C(K+1,I)=C(K,I)-B(K,K+1-I)*C(K+1,K+1)
      K=K+1
      IF (K.NE.NP) GOTO 42
      DO 62 I=1,NP
 62    H(I)=.5*B(NP,I)+.5*C(NP,I)
        WRITE(9,200)(H(I),I=1,NP)
200  FORMAT(6E15.7/6E15.7)
 21  CONTINUE
      RETURN
      END
C*************************************************************
      SUBROUTINE GAUSS(IX,S,AM,V)
C------------------------------------------------------------
C=> THIS ROUTINE PRODUCES GAUSSIAN DISTRIBUTED NOISE WITH
C     THE HELP OF THE ROUTINE 'RANDU'.
C------------------------------------------------------------
      A=0.0
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
 50  A=A+Y
      V=(A-6.0)*S+AM
      RETURN
      END
C*************************************************************
      SUBROUTINE RANDU(IX,IY,YFL)
      IY=IX*16807
      IF(IY)5,6,6
  5  IY=IY+2147483647
  6  YFL=IY
```

```
      YFL=YFL*.4656613E-9
      RETURN
      END
C****************************************************************
```

6. LPCs BASED ON SPEECH ENHANCED BY ANC

```
C*********************************************************
C=> THIS PROGRAM READS HEXADECIMAL SPEECH SAMPLES
C   FROM DEVICE#10, ENHANCES SPEECH USING ANC AND
C   CALCULATES LPC PARAMETERS WHICH ARE THEN STORED
C   ON DEVICE#9. SEE STATEMENT 100 OF
C   SUBROUTINE STORE FOR INPUT FORMAT AND 1200 AND
C   200 OF SUBROUTINE ANCLPC FOR OUTPUT FORMAT.
C*********************************************************
      REAL DATA(40000),X(40000),NTS
      INTEGER FLEN,DATA1(40000),SEED
      COMMON FRCVUV,TVUV,PRVFR
      PRVFR=.5
      SEED=65539
      NTS=0.
      FRCVUV=3.
      TVUV=2.
      TENRGY=9.
      NP=12
      MAX=40000
      FLEN=200
      CALL STORE(MAX,DATA,X,FLEN,TENRGY,NP,DATA1,SEED,
     &NTS)
      STOP
      END
C*********************************************************
      SUBROUTINE STORE(MAX,DATA,X,FLEN,TENRGY,NP,
     *DATA1,SEED,NTS)
C-------------------------------------------------------
C=> THIS SUBROUTINE READS IN THE RAW DATA, BRINGS
C     THEM TO 0 CENTER, (ASSUMING THAT THE RAW
C     DATA IS IN THE RANGE 0-FF HEX,) DETERMINES FIRST
C     AND LAST SPEECH FRAMES BASED ON THE THRESHOLD
C     'TENRGY'. IT, THEN, CALLS ROUTINE 'CHOP' THAT
C     ELIMINATES THE START AND END SILENT FRAMES.
C-------------------------------------------------------
      REAL DATA(MAX),X(MAX),NTS
      INTEGER FLEN,FRAME,FIRST,DATA1(MAX),SEED
C=> READ IN DATA SAMPLES AND SHIFT THEM TO 0 CENTER.
      READ(10,100,END=1)(DATA1(N),N=1,MAX)
  100 FORMAT(34Z2)
    1 N=N-1
      DO 9 I=1,N
    9 DATA(I)=DATA1(I)
      DO 2 I=1,N
    2  DATA(I)=DATA(I)-128.
```

```
      NF=N/FLEN
C=> DETERMINE FIRST AND LAST VOICED FRAME.
      DO 3 I=1,NF
        FRAME=I
        SS=0.
        DO 4 J=1,FLEN
    4     SS=SS+DATA(FLEN*(FRAME-1)+J)**2
        ENRGY=SS/FLEN
        IF (ENRGY.GE.TENRGY) GOTO 5
    3   CONTINUE
      PRINT,'NO VOICED FRAME FOUND W.R.T. THRESHOLD
     & ENERGY= ',TENRGY
      RETURN
    5 FIRST=FRAME
C=> FIRST IS THE NUMBER OF THE FIRST NON-SILENT
C     FRAME.
      DO 6 I=1,NF
        FRAME=NF-I+1
        SS=0.
        DO 7 J=1,FLEN
    7     SS=SS+DATA(FLEN*(FRAME-1)+J)**2
        ENRGY=SS/FLEN
        IF (ENRGY.GE.TENRGY) GOTO 8
    6 CONTINUE
    8 LAST=FRAME
C=> LAST IS THE NUMBER OF THE LAST NON-SILENT
C     FRAME.
      NOF=LAST-FIRST+1
      NOS=NOF*FLEN
      CALL CHOP(DATA,MAX,X,NOS,FIRST,FLEN,NOF,NP,SEED,
     &NTS)
      RETURN
      END
C*********************************************************
      SUBROUTINE CHOP(DATA,MAX,X,NOS,FIRST,FLEN,
     *NOF,NP,SEED,NTS)
C-------------------------------------------------------
C=> THIS SUBROUTINE ELIMINATES START AND END
C     SILENT FRAMES AND STORES THE REMAINING
C     SAMPLES IN ARRAY 'X'.
C-------------------------------------------------------
      REAL DATA(MAX),X(NOS),PHI(20,20),NTS
      INTEGER FIRST,FLEN,SEED
      DO 9 I=1,NOS
    9   X(I)=DATA(FLEN*(FIRST-1)+I)
      CALL ANCLPC(NOF,NOS,X,FLEN,NP,PHI,SEED,NTS)
      RETURN
      END
```

```
C*********************************************************
      SUBROUTINE ANCLPC(NOF,NOS,X,FLEN,NP,PHI,SEED,NTS)
C--------------------------------------------------------
C=> THIS SUBROUTINE ADDS NOISE TO SPEECH USING
C     SUBROUTINE 'GAUSS', ENHANCES SPEECH USING
C     'ANC' AND CALCULATES LPC PARAMETERS. THE LPC
C     PARAMETERS ARE STORED ON DEVICE#9. SEE
C     STATEMENTS 1200 AND 200 FOR OUTPUT FORMAT.
C--------------------------------------------------------
      COMMON FRCVUV,TVUV,PRVFR
      REAL RHO(212),PHI(NP,NP),X(NOS),NTS,NPHI,MEAN,
     &NB,NC,B(13,13),C(12,12),ALPHA(13),RSS(12),BPREV(13),
     &PHIR(200)
      INTEGER FRAME,FLEN,SEED,PLEN,FLENMK,PEAK,PP1,PPNP
C=>WRITE THE TOTAL NUMBER OF FRAMES.
      WRITE(9,1200)NOF
 1200 FORMAT(I3)
C=> FIND MEAN & SD. OF THE SPEECH DATA.
      IF (NTS.LT.1.E-4) GOTO 30
      SUM=SUMS=0.
      DO 19 I=1,NOS
        SUM=SUM+X(I)
   19   SUMS=SUMS+X(I)**2
      MEAN=SUM/NOS
      SD=SQRT((SUMS/NOS)-(MEAN**2))
C=> S IS THE STANDARD DEVIATION OF NOISE TO BE
C     ADDED TO SPEECH.
      S=NTS*SD/100.
C=> ADD NOISE TO SPEECH.
      DO 20 I=1,NOS
        CALL GAUSS(SEED,S,0.0,V)
   20   X(I)=X(I)+V
   30 IFLAG=0
      NPP1=NP+1
      DO 21 FRAME=1,NOF
        MAXP=0
C=> CALCULATE THE AUTOCORRELATION FUNCTION.
        SUMS=0.
        DO 22 I=1,FLEN
   22    SUMS=SUMS+X(FLEN*(FRAME-1)+I)**2
        ENRGY=SUMS/FLEN
C ENRGY IS ALSO = PHIR(0)
   33   DO 26 K=1,175
          FLENMK=FLEN-K
          PHIR(K)=0.
          DO 23 J=1,FLENMK
   23      PHIR(K)=PHIR(K)+X(FLEN*(FRAME-1)+J)*
     &       X(FLEN*(FRAME-1)+J+K)
```

```
26      PHIR(K)=PHIR(K)/FLEN
C=> CHECK FOR VOICED OR NOT AND ESTIMATE PITCH.
        GPMAX=0.
        IFV=0
        DO 24 PLEN=24,120
          N=FLEN/PLEN
          NM1=N-1
          IF(NM1.EQ.0)NM1=1
          GP=0.
          DO 25 L=1,NM1
25         GP=GP+PHIR(L*PLEN)
          GP=2.*PLEN*GP/FLEN
          IF(GP.GT.GPMAX)THEN
            GPMAX=GP
            MAXP=PLEN
            ENDIF
24     CONTINUE
        IF (GPMAX.LE.0.) GOTO 62
        VUV=FRCVUV*PHIR(MAXP)/ENRGY+GPMAX/ENRGY
        IF(VUV.GE.TVUV) IFV=1
62     IFLAG=IFLAG+1
C=> FIND THE ENHANCEMENT FILTER COEFFICIENTS.
      IF ((IFLAG.EQ.1).AND.(IFV.EQ.0)) THEN
        B(13,1)=1.
        DO 63 I=2,13
63       B(13,I)=0.
        GOTO 46
        ENDIF
      IF (IFV.EQ.0) GOTO 57
      PPNP=MAXP+NP
      DO 43 I=1,PPNP
43     RHO(I)=PHIR(I)/ENRGY
      DO 37 I=1,NPP1
        DO 37 J=1,NPP1
37       B(I,J)=0.
      DO 53 I=1,NP
        DO 53 J=1,NP
53       C(I,J)=0.
      B(1,1)=RHO(MAXP);C(1,1)=RHO(1)
      K=1
42   NB=RHO(MAXP+K)
      DB=1.
      DO 38 J=1,K
        NB=NB-RHO(J)*B(K,K+1-J)
38     DB=DB-RHO(J)*C(K,J)
      B(K+1,K+1)=NB/DB
      DO 39 I=1,K
39     B(K+1,I)=B(K,I)-C(K,K+1-I)*B(K+1,K+1)
```

```
      IF (K.EQ.NP) GOTO 46
      NC=RHO(K+1)
      DC=1.
      DO 40 J=1,K
        NC=NC-RHO(J)*C(K,K+1-J)
   40   DC=DC-RHO(J)*C(K,J)
      C(K+1,K+1)=NC/DC
      DO 41 I=1,K
   41 C(K+1,I)=C(K,I)-C(K,K+1-I)*C(K+1,K+1)
      K=K+1
      GOTO 42
   46 IF (IFLAG.EQ.1) GOTO 60
      DO 61 I=1,13
   61   B(13,I)=PRVFR*BPREV(I)+(1.-PRVFR)*B(13,I)
   60 DO 55 I=1,13
   55   BPREV(I)=B(13,I)
C=> NOW B(13,I),I=1,...,13 HAVE BEEN OBTAINED.
C     FIND ALPHA(I),I=1,...,13.
      DO 49 I=1,13
        NPMIP2=NP-I+2
        ALPHA(I)=0.
        DO 49 L=1,NPMIP2
   49     ALPHA(I)=ALPHA(I)+B(13,L)*B(13,L+I-1)
C=> ESTIMATE AUTOCORRELATION FUNCTION OF THE
C      ENHANCED SPEECH.
   57 ENRGSS=ALPHA(1)*ENRGY
      DO 50 I=2,NPP1
   50   ENRGSS=ENRGSS+2.*ALPHA(I)*PHIR(I-1)
      DO 48 J=1,NP
        JP2=J+2
        RSS(J)=ALPHA(J+1)*ENRGY
        DO 52 I=2,NPP1
   52     RSS(J)=RSS(J)+ALPHA(I)*PHIR(J+I-1)
        DO 51 I=1,J
   51     RSS(J)=RSS(J)+ALPHA(I)*PHIR(J-I+1)
        IF (J.EQ.NP) GOTO 48
        DO 54 I=JP2,NPP1
   54     RSS(J)=RSS(J)+ALPHA(I)*PHIR(I-J-1)
   48   CONTINUE
C=> CALCULATE THE LPC PARAMETERS AND STORE ON
C      DEVICE#9.
   34   DO 12 I=1,NP
          RHO(I)=0.
         DO 12 J=1,NP
   12      PHI(I,J)=0.
        DO 16 I=1,NP
   16     RHO(I)=RSS(I)/ENRGSS
        PHI(1,1)=RHO(1)
```

```
        DO 17 I=2,NP
          NPHI=RHO(I)
          DPHI=1.
          IM1=I-1
          DO 18 J=1,IM1
            NPHI=NPHI-PHI(I-1,J)*RHO(I-J)
   18       DPHI=DPHI-PHI(I-1,J)*RHO(J)
          PHI(I,I)=NPHI/DPHI
          DO 17 J=1,IM1
   17       PHI(I,J)=PHI(I-1,J)-PHI(I,I)*PHI(I-1,I-J)
C WRITING OUTPUT IN PROPER FILES
        WRITE(9,200)(PHI(NP,I),I=1,NP)
  200 FORMAT(6E15.7/6E15.7)
   21 CONTINUE
      RETURN
      END
C*****************************************************
      SUBROUTINE GAUSS(IX,S,AM,V)
C-----------------------------------------------------
C=> THIS ROUTINE PRODUCES GAUSSIAN DISTRIBUTED NOISE WITH
C     THE HELP OF THE ROUTINE 'RANDU'.
C-----------------------------------------------------
      A=0.0
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
   50 A=A+Y
      V=(A-6.0)*S+AM
      RETURN
      END
C*****************************************************
      SUBROUTINE RANDU(IX,IY,YFL)
      IY=IX*16807
      IF(IY)5,6,6
    5 IY=IY+2147483647
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
C*****************************************************
```

7. LPCs BASED ON SPEECH ENHANCED BY LPS

```
C******************************************************
C=> THIS PROGRAM READS HEXADECIMAL SPEECH SAMPLES
C   FROM DEVICE#10, ENHANCES SPEECH USING LPS AND
C   CALCULATES LPC PARAMETERS WHICH ARE THEN STORED
C   ON DEVICE#9. SEE STATEMENT 100 OF
C   SUBROUTINE 'STORE' FOR INPUT FORMAT AND 1200 AND
C   200 OF SUBROUTINE 'LPSLPC' FOR OUTPUT FORMAT.
C******************************************************
      REAL DATA(40000),X(40000),NTS
      INTEGER FLEN,DATA1(40000),SEED
      COMMON PRVFR
      PRVFR=.5
      SEED=65539
      NTS=0.
      TENRGY=9.0
      NP=12
      MAX=40000
      FLEN=200
      CALL STORE(MAX,DATA,X,FLEN,TENRGY,NP,DATA1,SEED,
     &NTS)
      STOP
      END
C******************************************************
      SUBROUTINE STORE(MAX,DATA,X,FLEN,TENRGY,NP,
     *DATA1,SEED,NTS)
C-----------------------------------------------------
C=> THIS SUBROUTINE READS IN THE RAW DATA, BRINGS
C     THEM TO 0 CENTER, (ASSUMING THAT THE RAW
C     DATA IS IN THE RANGE 0-FF HEX,) DETERMINES FIRST
C     AND LAST SPEECH FRAMES BASED ON THE THRESHOLD
C     'TENRGY'. IT, THEN, CALLS ROUTINE 'CHOP' THAT
C     ELIMINATES THE START AND END SILENT FRAMES.
C-----------------------------------------------------
      REAL DATA(MAX),X(MAX),NTS
      INTEGER FLEN,FRAME,FIRST,DATA1(MAX),SEED
C=> READ IN DATA SAMPLES AND SHIFT THEM TO 0 CENTER.
      READ(10,100,END=1)(DATA1(N),N=1,MAX)
  100 FORMAT(34Z2).
    1 N=N-1
      DO 9 I=1,N
    9 DATA(I)=DATA1(I)
      DO 2 I=1,N
    2  DATA(I)=DATA(I)-128.
      NF=N/FLEN
C=> DETERMINE FIRST AND LAST VOICED FRAME.
```

```
      DO 3 I=1,NF
        FRAME=I
        SS=0.
        DO 4 J=1,FLEN
    4     SS=SS+DATA(FLEN*(FRAME-1)+J)**2
        ENRGY=SS/FLEN
        IF (ENRGY.GE.TENRGY) GOTO 5
    3   CONTINUE
      PRINT,'NO VOICED FRAME FOUND W.R.T. THRESHOLD
     & ENERGY=',TENRGY
      RETURN
    5 FIRST=FRAME
C=> FIRST IS THE NUMBER OF THE FIRST NON-SILENT
C     FRAME.
      DO 6 I=1,NF
        FRAME=NF-I+1
        SS=0.
        DO 7 J=1,FLEN
    7     SS=SS+DATA(FLEN*(FRAME-1)+J)**2
        ENRGY=SS/FLEN
        IF (ENRGY.GE.TENRGY) GOTO 8
    6 CONTINUE
    8 LAST=FRAME
C=> LAST IS THE NUMBER OF THE LAST NON-SILENT
C     FRAME.
      NOF=LAST-FIRST+1
      NOS=NOF*FLEN
      CALL CHOP(DATA,MAX,X,NOS,FIRST,FLEN,NOF,NP,SEED,
     &NTS)
      RETURN
      END
C*****************************************************
      SUBROUTINE CHOP(DATA,MAX,X,NOS,FIRST,FLEN,
     *NOF,NP,SEED,NTS)
C-----------------------------------------------------
C=> THIS SUBROUTINE ELIMINATES START AND END
C     SILENT FRAMES AND STORES THE REMAINING
C     SAMPLES IN ARRAY 'X'.
C-----------------------------------------------------
      REAL DATA(MAX),X(NOS),PHI(20,20),NTS
      INTEGER FIRST,FLEN,SEED
      DO 9 I=1,NOS
    9   X(I)=DATA(FLEN*(FIRST-1)+I)
      CALL LPSLPC(NOF,NOS,X,FLEN,NP,PHI,SEED,NTS)
      RETURN
      END
```

```
C*****************************************************
      SUBROUTINE LPSLPC(NOF,NOS,X,FLEN,NP,PHI,SEED,
     *NTS)
C-----------------------------------------------------
C=> THIS SUBROUTINE ADDS NOISE TO SPEECH USING
C     SUBROUTINE 'GAUSS', ENHANCES SPEECH USING
C     'LPS' AND CALCULATES LPC PARAMETERS. THE LPC
C     PARAMETERS ARE STORED ON DEVICE#9. SEE
C     STATEMENTS 1200 AND 200 FOR OUTPUT FORMAT.
C-----------------------------------------------------
      COMMON PRVFR
      REAL RHO(20),PHI(NP,NP),X(NOS),NTS,NPHI,MEAN,
     &PHIR(24),NB,NC,B(13,13),ALPHA(13),RSS(12),BPREV(13)
      INTEGER FRAME,FLEN,SEED,FLENMK
C=> WRITE THE TOTAL NUMBER OF FRAMES.
      WRITE(9,1200)NOF
 1200 FORMAT(I3)
C=> FIND MEAN & SD. OF THE SPEECH DATA.
      S=0.
      IF (NTS.LT.1.) GOTO 30
      SUM=SUMS=0.
      DO 19 I=1,NOS
        SUM=SUM+X(I)
   19   SUMS=SUMS+X(I)**2
      MEAN=SUM/NOS
      SD=SQRT((SUMS/NOS)-(MEAN**2))
C=> S IS THE STANDARD DEVIATION OF NOISE TO BE
C     ADDED TO SPEECH.
      S=NTS*SD/100.
C=> ADD NOISE TO SPEECH.
      DO 20 I=1,NOS
        CALL GAUSS(SEED,S,0.0,V)
   20   X(I)=X(I)+V
   30 NPT2=NP*2
      BPREV(1)=1.
      DO 59 I=2,13
   59   BPREV(I)=0.
      DO 21 FRAME=1,NOF
C=> CALCULATE THE AUTOCORRELATION FUNCTION.
        SUMS=0.
        DO 22 I=1,FLEN
   22     SUMS=SUMS+X(FLEN*(FRAME-1)+I)**2
        ENRGY=SUMS/FLEN
C=> ENRGY IS ALSO = PHIR(0)
   33   DO 26 K=1,NPT2
          FLENMK=FLEN-K
          PHIR(K)=0.
          DO 23 J=1,FLENMK
```

```
   23       PHIR(K)=PHIR(K)+X(FLEN*(FRAME-1)+J)*
     &        X(FLEN*(FRAME-1)+J+K)
   26     PHIR(K)=PHIR(K)/FLEN
C=> GET B(13,I),I=1,2,..,13.
      NPP1=NP+1
      DO 43 I=1,NPP1
   43   RHO(I)=PHIR(I)/ENRGY
      DO 37 I=1,NPP1
        DO 37 J=1,NPP1
   37     B(I,J)=0.
      B(1,1)=RHO(1)
      K=1
   42 NB=RHO(K+1)
      DB=1.
      DO 38 J=1,K
        NB=NB-RHO(J)*B(K,K+1-J)
   38   DB=DB-RHO(J)*B(K,J)
      B(K+1,K+1)=NB/DB
      DO 39 I=1,K
   39   B(K+1,I)=B(K,I)-B(K,K+1-I)*B(K+1,K+1)
      IF (K.EQ.NP) GOTO 46
      K=K+1
      GOTO 42
   46 DO 61 I=1,13
   61   B(13,I)=PRVFR*BPREV(I)+(1.-PRVFR)*B(13,I)
      DO 55 I=1,13
   55   BPREV(I)=B(13,I)
C=> NOW B(13,I),I=1,...,13 HAVE BEEN OBTAINED.
C     FIND ALPHA(I),I=1,...,13.
      DO 49 I=1,13
        NPMIP2=NP-I+2
        ALPHA(I)=0.
        DO 49 L=1,NPMIP2
   49     ALPHA(I)=ALPHA(I)+B(13,L)*B(13,L+I-1)
   57 ENRGSS=ALPHA(1)*ENRGY
      DO 50 I=2,NPP1
   50   ENRGSS=ENRGSS+2.*ALPHA(I)*PHIR(I-1)
      DO 48 J=1,NP
        JP2=J+2
        RSS(J)=ALPHA(J+1)*ENRGY
        DO 52 I=2,NPP1
   52     RSS(J)=RSS(J)+ALPHA(I)*PHIR(J+I-1)
        DO 51 I=1,J
   51     RSS(J)=RSS(J)+ALPHA(I)*PHIR(J-I+1)
        IF (J.EQ.NP) GOTO 48
        DO 54 I=JP2,NPP1
   54     RSS(J)=RSS(J)+ALPHA(I)*PHIR(I-J-1)
   48   CONTINUE
```

```
   34   DO 12 I=1,NP
          RHO(I)=0.
         DO 12 J=1,NP
   12      PHI(I,J)=0.
        DO 16 I=1,NP
   16    RHO(I)=RSS(I)/ENRGSS
        PHI(1,1)=RHO(1)
        DO 17 I=2,NP
          NPHI=RHO(I)
          DPHI=1.
          IM1=I-1
          DO 18 J=1,IM1
            NPHI=NPHI-PHI(I-1,J)*RHO(I-J)
   18      DPHI=DPHI-PHI(I-1,J)*RHO(J)
          PHI(I,I)=NPHI/DPHI
          DO 17 J=1,IM1
   17      PHI(I,J)=PHI(I-1,J)-PHI(I,I)*PHI(I-1,I-J)
C WRITING OUTPUT IN PROPER FILES
        WRITE(9,200)(PHI(NP,I),I=1,NP)
  200 FORMAT(6E15.7/6E15.7)
   21 CONTINUE
      RETURN
      END
C*****************************************************************
      SUBROUTINE GAUSS(IX,S,AM,V)
C-----------------------------------------------------------------
C=> THIS ROUTINE PRODUCES GAUSSIAN DISTRIBUTED NOISE WITH
C     THE HELP OF THE ROUTINE 'RANDU'.
C-----------------------------------------------------------------
      A=0.0
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
   50 A=A+Y
      V=(A-6.0)*S+AM
      RETURN
      END
C*****************************************************************
      SUBROUTINE RANDU(IX,IY,YFL)
      IY=IX*16807
      IF(IY)5,6,6
    5 IY=IY+2147483647
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
C*****************************************************************
```

8. LPCs BASED ON SPEECH ENHANCED BY AFT

```
C**********************************************************
C=> THIS PROGRAM READS HEXADECIMAL SPEECH SAMPLES
C   FROM DEVICE#10, ENHANCES SPEECH USING AFT AND
C   CALCULATES LPC PARAMETERS WHICH ARE THEN STORED
C   ON DEVICE#9. SEE STATEMENT 100 OF
C   SUBROUTINE STORE FOR INPUT FORMAT AND 1200 AND
C   200 OF SUBROUTINE AFTLPC FOR OUTPUT FORMAT.
C**********************************************************
      REAL DATA(40000),X(40000),NTS
      INTEGER FLEN,DATA1(40000),SEED
      COMMON PRVFR
      PRVFR=.5
      SEED=65539
      NTS=0.
      TENRGY=9.0
      NP=12
      MAX=40000
      FLEN=200
      CALL STORE(MAX,DATA,X,FLEN,TENRGY,NP,DATA1,
     *SEED,NTS)
      STOP
      END
C**********************************************************
      SUBROUTINE STORE(MAX,DATA,X,FLEN,TENRGY,NP,
     *DATA1,SEED,NTS)
C----------------------------------------------------------
C=> THIS SUBROUTINE READS IN THE RAW DATA, BRINGS
C     THEM TO 0 CENTER, (ASSUMING THAT THE RAW
C     DATA IS IN THE RANGE 0-FF HEX,) DETERMINES FIRST
C     AND LAST SPEECH FRAMES BASED ON THE THRESHOLD
C     'TENRGY'. IT, THEN, CALLS ROUTINE 'CHOP' THAT
C     ELIMINATES THE START AND END SILENT FRAMES.
C----------------------------------------------------------
      REAL DATA(MAX),X(MAX),NTS
      INTEGER FLEN,FRAME,FIRST,DATA1(MAX),SEED
C=> READ IN DATA SAMPLES AND SHIFT THEM TO 0 CENTER.
      READ(10,100,END=1)(DATA1(N),N=1,MAX)
  100 FORMAT(34Z2)
    1 N=N-1
      DO 9 I=1,N
    9 DATA(I)=DATA1(I)
      DO 2 I=1,N
    2  DATA(I)=DATA(I)-128.
```

```
      NF=N/FLEN
C=> DETERMINE FIRST AND LAST VOICED FRAME.
      DO 3 I=1,NF
         FRAME=I
         SS=0.
         DO 4 J=1,FLEN
    4     SS=SS+DATA(FLEN*(FRAME-1)+J)**2
         ENRGY=SS/FLEN
         IF (ENRGY.GE.TENRGY) GOTO 5
    3   CONTINUE
      PRINT,'NO VOICED FRAME FOUND W.R.T.
     *THRESHOLD ENERGY=',TENRGY
      RETURN
    5 FIRST=FRAME
C=> FIRST IS THE NUMBER OF THE FIRST NON-SILENT
C     FRAME.
      DO 6 I=1,NF
         FRAME=NF-I+1
         SS=0.
         DO 7 J=1,FLEN
    7     SS=SS+DATA(FLEN*(FRAME-1)+J)**2
         ENRGY=SS/FLEN
         IF (ENRGY.GE.TENRGY) GOTO 8
    6 CONTINUE
    8 LAST=FRAME
C=> LAST IS THE NUMBER OF THE LAST NON-SILENT
C     FRAME.
      NOF=LAST-FIRST+1
      NOS=NOF*FLEN
      CALL CHOP(DATA,MAX,X,NOS,FIRST,FLEN,NOF,
     *NP,SEED,NTS)
      RETURN
      END
C****************************************************
      SUBROUTINE CHOP(DATA,MAX,X,NOS,FIRST,FLEN,
     *NOF,NP,SEED,NTS)
C----------------------------------------------------
C=> THIS SUBROUTINE ELIMINATES START AND END
C     SILENT FRAMES AND STORES THE REMAINING
C     SAMPLES IN ARRAY 'X'.
C----------------------------------------------------
      REAL DATA(MAX),X(NOS),PHI(20,20),NTS
      INTEGER FIRST,FLEN,SEED
      DO 9 I=1,NOS
    9  X(I)=DATA(FLEN*(FIRST-1)+I)
      CALL AFTLPC(NOF,NOS,X,FLEN,NP,PHI,SEED,NTS)
      RETURN
      END
```

```
C*****************************************************
      SUBROUTINE AFTLPC(NOF,NOS,X,FLEN,NP,PHI,SEED,
     *NTS)
C-----------------------------------------------------
C=> THIS SUBROUTINE ADDS NOISE TO SPEECH USING
C     SUBROUTINE 'GAUSS', ENHANCES SPEECH USING
C     'AFT' AND CALCULATES LPC PARAMETERS. THE LPC
C     PARAMETERS ARE STORED ON DEVICE#9. SEE
C     STATEMENTS 1200 AND 200 FOR OUTPUT FORMAT.
C-----------------------------------------------------
      COMMON PRVFR
      REAL RHO(20),PHI(NP,NP),X(NOS),NTS,NPHI,
     &MEAN,PHIR(24),NB,NC,
     &B(13,13),C(12,12),ALPHA(13),RSS(12),BPREV(13)
      INTEGER FRAME,FLEN,SEED,FLENMK
C=>WRITE THE TOTAL NUMBER OF FRAMES.
      WRITE(9,1200)NOF
 1200 FORMAT(I3)
C=> FIND MEAN & SD. OF THE SPEECH DATA.
      S=0.
      IF (NTS.LT.1.) GOTO 30
      SUM=SUMS=0.
      DO 19 I=1,NOS
        SUM=SUM+X(I)
   19   SUMS=SUMS+X(I)**2
      MEAN=SUM/NOS
      SD=SQRT((SUMS/NOS)-(MEAN**2))
C=> 'S' IS THE STANDARD DEVIATION OF NOISE TO BE
C     ADDED TO SPEECH.
      S=NTS*SD/100.
C=> ADD NOISE TO SPEECH.
      DO 20 I=1,NOS
        CALL GAUSS(SEED,S,0.0,V)
   20   X(I)=X(I)+V
   30 NPT2=NP*2
      BPREV(1)=1.
      DO 59 I=2,13
   59   BPREV(I)=0.
      DO 21 FRAME=1,NOF
C=> CALCULATE THE AUTOCORRELATION FUNCTION.
        SUMS=0.
        DO 22 I=1,FLEN
   22     SUMS=SUMS+X(FLEN*(FRAME-1)+I)**2
        ENRGY=SUMS/FLEN
C=> ENRGY IS ALSO = PHIR(0)
   33   DO 26 K=1,NPT2
          FLENMK=FLEN-K
          PHIR(K)=0.
```

```
          DO 23 J=1,FLENMK
   23       PHIR(K)=PHIR(K)+X(FLEN*(FRAME-1)+J)*
     &        X(FLEN*(FRAME-1)+J+K)
   26     PHIR(K)=PHIR(K)/FLEN
C=> FIND THE ENHANCEMENT FILTER COEFFS.
C     B(NP+1,I), I=1,2,..,NP+1.
      NPP1=NP+1
      DO 43 I=1,NP
   43   RHO(I)=PHIR(I)/ENRGY
      DO 37 I=1,NPP1
        DO 37 J=1,NPP1
   37     B(I,J)=0.
      DO 53 I=1,NP
        DO 53 J=1,NP
   53     C(I,J)=0.
      B(1,1)=1.-(S**2/ENRGY);C(1,1)=RHO(1)
      K=1
   42 NB=RHO(K)
      DB=1.
      DO 38 J=1,K
        NB=NB-RHO(J)*B(K,K+1-J)
   38   DB=DB-RHO(J)*C(K,J)
      B(K+1,K+1)=NB/DB
      DO 39 I=1,K
   39   B(K+1,I)=B(K,I)-C(K,K+1-I)*B(K+1,K+1)
      IF (K.EQ.NP) GOTO 46
      NC=RHO(K+1)
      DC=1.
      DO 40 J=1,K
        NC=NC-RHO(J)*C(K,K+1-J)
   40   DC=DC-RHO(J)*C(K,J)
      C(K+1,K+1)=NC/DC
      DO 41 I=1,K
   41 C(K+1,I)=C(K,I)-C(K,K+1-I)*C(K+1,K+1)
      K=K+1
      GOTO 42
C=> SMOOTH OUT FILTER COEFFICIENTS.
   46 DO 61 I=1,13
   61   B(13,I)=PRVFR*BPREV(I)+(1.-PRVFR)*B(13,I)
      DO 55 I=1,13
   55   BPREV(I)=B(13,I)
C=> ESTIMATE AUTOCORRELATION FUNCTION OF THE
C     ENHANCED SPEECH.
      DO 49 I=1,13
        NPMIP2=NP-I+2
        ALPHA(I)=0.
        DO 49 L=1,NPMIP2
   49     ALPHA(I)=ALPHA(I)+B(13,L)*B(13,L+I-1)
```

```
   57 ENRGSS=ALPHA(1)*ENRGY
      DO 50 I=2,NPP1
   50   ENRGSS=ENRGSS+2.*ALPHA(I)*PHIR(I-1)
      DO 48 J=1,NP
        JP2=J+2
        RSS(J)=ALPHA(J+1)*ENRGY
        DO 52 I=2,NPP1
   52     RSS(J)=RSS(J)+ALPHA(I)*PHIR(J+I-1)
        DO 51 I=1,J
   51     RSS(J)=RSS(J)+ALPHA(I)*PHIR(J-I+1)
        IF (J.EQ.NP) GOTO 48
        DO 54 I=JP2,NPP1
   54     RSS(J)=RSS(J)+ALPHA(I)*PHIR(I-J-1)
   48   CONTINUE
C=> CALCULATE THE LPC PARAMETERS AND STORE ON
C       DEVICE#9.
   34   DO 12 I=1,NP
           RHO(I)=0.
          DO 12 J=1,NP
   12       PHI(I,J)=0.
        DO 16 I=1,NP
   16     RHO(I)=RSS(I)/ENRGSS
        PHI(1,1)=RHO(1)
        DO 17 I=2,NP
          NPHI=RHO(I)
          DPHI=1.
          IM1=I-1
          DO 18 J=1,IM1
            NPHI=NPHI-PHI(I-1,J)*RHO(I-J)
   18       DPHI=DPHI-PHI(I-1,J)*RHO(J)
          PHI(I,I)=NPHI/DPHI
          DO 17 J=1,IM1
   17       PHI(I,J)=PHI(I-1,J)-PHI(I,I)*PHI(I-1,I-J)
        WRITE(9,200)(PHI(NP,I),I=1,NP)
  200 FORMAT(6E15.7/6E15.7)
   21 CONTINUE
      RETURN
      END
C*********************************************************
      SUBROUTINE GAUSS(IX,S,AM,V)
C--------------------------------------------------------
C=> THIS ROUTINE PRODUCES GAUSSIAN DISTRIBUTED NOISE WITH
C     THE HELP OF THE ROUTINE 'RANDU'.
C--------------------------------------------------------
      A=0.0
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
```

```
   50 A=A+Y
      V=(A-6.0)*S+AM
      RETURN
      END
C*********************************************************
      SUBROUTINE RANDU(IX,IY,YFL)
      IY=IX*16807
      IF(IY)5,6,6
    5 IY=IY+2147483647
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
C*********************************************************
```

9. *REFERENCE TEMPLATE CREATION*

```
C*******************************************************
C=> THIS PROGRAM CREATES THE REFERENCE TEMPLATES FROM THE
C    LPC PARAMETERS OBTAINED USING THE PREVIOUSLY LISTED
C    PROGRAMS.
C=> THE INPUT LPC PARAMETER FILES SHOULD HAVE DEVICE
C    NUMBERS FROM 11 TO 10+NUTT, WHERE 'NUTT' IS THE
C    NUMBER OF UTTERANCES USED TO CREATE THE REFERENCE
C    TEMPLATES.
C=> THE OUTPUT IS WRITTEN TO THE FILE DEFINED AS
C    DEVICE NUMBER 10. PARAMETERS ARE STORED IN THE
C    REFERENCE TEMPLATES ACCORDING TO THE FOLLOWING
C    FORMAT:
C-------------------------------------------------------
C      PARAMETER:              LINE#:      FORMAT:
C     ------------           --------    ---------
C # OF UTTERANCES
C & TOTAL # OF FRAMES.......    1          (I2,I4)
C EIGENVALUES (NP)..........   3-4     (6E15.7/6E15.7)
C EIGENVECTORS (NPXNP)......   6-29    (6E15.7/6E15.7)
C (COLUMNS OF THIS MATRIX ARE EIGENVECTORS)
C ORTHOGONAL
C LPC PARAMETERS (NP).......  31-32    (6E15.7/6E15.7)
C-------------------------------------------------------
C=> EXTERNAL SUBROUTINES REQUIRED:
C    SUBROUTINE 'EIGEN' OF THE 'SSP' LIBRARY.
C*******************************************************
      DIMENSION R(20,20),XMEAN(20),RMEAN(20,20),B(20,20),
     &          PHI(20)
      NP=12
      NUTT=5
      CALL MAIN(NP,R,XMEAN,NUTT,RMEAN,B,PHI)
      STOP
      END
C*******************************************************
      SUBROUTINE MAIN(NP,R,XMEAN,NUTT,RMEAN,B,PHI)
C-------------------------------------------------------
C=> THIS SUBROUTINE READS IN THE LPC PARAMETERS AND
C    CALLS THE ROUTINE 'COMAT' THAT CALCULATES THE
C    COVARIANCE MATRICES OF THE LPC PARAMETERS. THE
C    COVARIANCE MATRICES ARE, THEN, AVERAGED OVER ALL
C    REFERENCE UTTERANCES. THE SUBROUTINE 'EIGEN' IS
C    CALLED FROM THE 'SSP' LIBRARY THAT CALCULATES THE
C    EIGENVALUES AND EIGENVECTORS OF THE MEAN COVARIANCE
C    MATRIX. THEN, THE SUBROUTINE 'ORTHO' IS CALLED
C    THAT CALCULATES THE ORTHOGONAL LPC PARAMETERS.
```

```
C-----------------------------------------------------------
      DIMENSION X(20,800),R(NP,NP),XMEAN(NP),RMEAN(NP,NP),
     &B(NP,NP),PHI(NP),DUMMY(78),BT(144)
      NNOF=0
      DO 6 I=1,NP
        DO 6 J=1,NP
 6        RMEAN(I,J)=0.
      DO 7 IUTT=1,NUTT
        IDEV=IUTT+10
        READ(IDEV,100)NOF
 100    FORMAT(I3)
        CALL COMAT(IDEV,NOF,NP,X,R,XMEAN)
        DO 8 I=1,NP
          DO 8 J=1,I
 8          RMEAN(I,J)=RMEAN(I,J)+NOF*R(I,J)
 7        NNOF=NNOF+NOF
        DO 9 I=1,NP
          DO 9 J=1,I
            RMEAN(I,J)=RMEAN(I,J)/NNOF
 9          RMEAN(J,I)=RMEAN(I,J)
        K=0
        DO 11 J=1,NP
          DO 11 I=1,J
            K=K+1
 11         DUMMY(K)=RMEAN(I,J)
        CALL EIGEN(DUMMY,BT,NP,0)
        L=0
        K=0
        DO 13 J=1,NP
          DO 12 I=1,NP
            K=K+1
 12         B(I,J)=BT(K)
          DO 13 I=1,J
            L=L+1
 13         RMEAN(I,J)=DUMMY(L)
C=> WRITE THE NUMBER OF UTTERANCES, TOTAL NUMBER OF
C    FRAMES, EIGENVALUES AND EIGENVECTORS IN THE OUTPUT
C    FILE.
        WRITE(10,300)NUTT,NNOF,(RMEAN(I,I),I=1,NP)
 300    FORMAT(I2,I4,//,6E15.7/6E15.7,/)
        DO 10 I=1,NP
 10      WRITE(10,400)(B(I,J),J=1,NP)
 400    FORMAT(6E15.7/6E15.7)
        CALL ORTHO(NUTT,NP,PHI,B)
        RETURN
        END
```

```
C*******************************************************
        SUBROUTINE COMAT(IDEV,NOF,NP,X,R,XMEAN)
        DIMENSION X(NP,NOF),R(NP,NP),XMEAN(NP)
C CALCULATION OF COVARIANCE MATRIX R
      DO 1 I=1,NP
        XMEAN(I)=0.
        DO 1 J=1,NP
    1     R(I,J)=0.
      DO 7 J=1,NOF
        READ(IDEV,200,END=6)(X(I,J),I=1,NP)
  200  FORMAT(6E15.7/6E15.7)
        NOFA=J
    7  CONTINUE
    6 NOF=NOFA
      REWIND IDEV
      DO 2 J=1,NOF
        DO 2 I=1,NP
    2     XMEAN(I)=XMEAN(I)+X(I,J)
      DO 3 I=1,NP
    3  XMEAN(I)=XMEAN(I)/NOF
      DO 4 I=1,NP
        DO 4 K=1,I
          DO 5 J=1,NOF
    5       R(I,K)=R(I,K)+(X(I,J)-XMEAN(I))*
     &(X(K,J)-XMEAN(K))
    4     R(I,K)=R(I,K)/(NOF-1)
      RETURN
      END
C*******************************************************
      SUBROUTINE ORTHO(NUTT,NP,PHI,B)
      DIMENSION PHI(NP),B(NP,NP),X(20,800)
C=> INITIALIZATION
      DO 12 I=1,NP
   12  PHI(I)=0.
      NNOF=0
      DO 11 IUTT=1,NUTT
        IDEV=IUTT+10
        READ(IDEV,100)NOF
  100  FORMAT(I3)
        CALL APHI(NP,NOF,X,IDEV,PHI,B)
   11  NNOF=NNOF+NOF
      DO 13 I=1,NP
   13  PHI(I)=PHI(I)/NNOF
      WRITE(10,200)(PHI(I),I=1,NP)
  200 FORMAT(/,6E15.7/6E15.7)
      RETURN
      END
```

```
C*****************************************************************
      SUBROUTINE APHI(NP,NOF,X,IDEV,PHI,B)
      DIMENSION X(NP,NOF),PHI(NP),B(NP,NP)
      DO 15 J=1,NOF
        READ(IDEV,500,END=16)(X(I,J),I=1,NP)
   15   NOFA=J
   16 NOF=NOFA
      DO 14 J=1,NOF
        DO 14 I=1,NP
          DO 14 K=1,NP
   14       PHI(I)=PHI(I)+B(K,I)*X(K,J)
  500 FORMAT(6E15.7/6E15.7)
      RETURN
      END
C*****************************************************************
```

```
C*********************************************************
C=> THIS PROGRAM IDENTIFIES THE TEST SPEAKER BASED UPON
C    THE REFERENCE TEMPLATES CREATED BY REFERENCE
C    TEMPLATE CREATING PROGRAM LISTED PREVIOUSLY.
C=> THE INPUT REFERENCE TEMPLATE FILES SHOULD HAVE
C    DEVICE NUMBERS 11 TO 10+NTEST, WHERE 'NTEST' IS
C    THE SIZE OF THE CUSTOMER (OR REFERENCE) SET.
C=> THE INPUT TEST FILE SHOULD HAVE DEVICE NUMBER 10,
C    AND SHOULD CONTAIN THE NUMBER OF FRAMES AND
C    LPC PARAMETERS IN THE APPROPRIATE FORMAT AS USED
C    BY THE LPC ESTIMATION PROGRAMS INCLUDED IN THIS
C    APPENDIX.
C---------------------------------------------------------
C=> NAMES: THE ARRAY 'NAMES' SHOULD HAVE A DIMENSION
C    EQUAL TO 'NTEST' AND SHOULD CONTAIN THE NAMES OF
C    THE REFERENCE SPEAKERS (UPTO EIGHT CHARACTERS
C    IN LENGTH) IN THE SAME ORDER AS
C    THEIR REFERENCE TEMPLATE FILES (I.E. NAMES(I)
C    SHOULD BE THE NAME OF THE SPEAKER WHOSE REFERENCE
C    TEMPLATE IS DEFINED BY DEVICE NUMBER 10+I).
C---------------------------------------------------------
C*********************************************************
      CHARACTER*8 NAMES(11)
      DATA NAMES/----,----,------/
      NTEST=11
      NP=12
      NOP=0
      CALL TEST(NTEST,NP,NOP,NAMES)
      STOP
      END
C*********************************************************
      SUBROUTINE TEST(NTEST,NP,NOP,NAMES)
      CHARACTER*8 NAMES(NTEST)
      DIMENSION PHI(20),B(20,20),X(20,200),Z(20),D(11)
      REAL LAMDA(20)
      READ(10,100)NOF
  100 FORMAT(I3)
      CALL ORTHO(NOF,NP,NTEST,LAMDA,B,Z,NOP,D,X,PHI)
      N=1
      AMIN=D(1)
      DO 1 I=2,NTEST
         J=I
         IF (D(J).LT.AMIN) THEN
            AMIN=D(J)
            N=J
```

```
       ENDIF
    1   CONTINUE
       WRITE(6,600)NAMES(N)
  600 FORMAT('0',' MOST PROBABLY THE GIVEN SPEECH IS OF
      & THE PERSON NAMED ',A8,/)
       WRITE(6,800)
  800 FORMAT('0',4X,'****THE CALCULATED DISTANCES ARE
      &****'//,7X,'NAME'
      &,16X,'DISTANCE'/,7X,'----',16X,'--------',/)
       DO 2 I=1,NTEST
    2 WRITE(6,900)NAMES(I),D(I)
  900 FORMAT(' ',5X,A8,12X,F12.2,/)
       WRITE(6,700)
  700 FORMAT(80('*'))
       RETURN
       END
C*************************************************************
       SUBROUTINE ORTHO(NOF,NP,NTEST,LAMDA,B,Z,NOP,D,X,
      &PHI)
C-------------------------------------------------------------
C=> THIS SUBROUTINE READS IN THE LPC PARAMETERS OF
      THE UNKNOWN SPEAKER AND CALCULATES THE OLPC
      PARAMETERS USING THE REFERENCE TEMPLATES OF EACH
      OF THE REFERENCE SPEAKERS. IT, THEN, FINDS THE
      REQUIRED RECOGNITION DISTANCES.
C-------------------------------------------------------------
       DIMENSION X(NP,NOF),PHI(NP),B(NP,NP),Z(NP),D(NTEST)
       REAL LAMDA(NP)
       NPD=NOP+1
       NOF=NOF+3
C=> READ IN THE LPC PARAMETERS OF THE UNKNOWN SPEAKER.
       DO 1 J=1,NOF
    1   READ(10,200,END=8)(X(I,J),I=1,NP)
  200 FORMAT(6E15.7/6E15.7)
    8 NOF=J-1
       DO 3 M=1,NTEST
         D(M)=0.
         IDEV=M+10
         DO 4 I=1,NP
    4     Z(I)=0.
C=> READ IN THE REFERENCE TEMPLATE OF REFERENCE SPEAKER
C    NUMBER M.
         READ(IDEV,300)NUTT,NNOF,(LAMDA(I),I=1,NP)
  300   FORMAT(I2,I4//6E15.7/6E15.7/)
         DO 2 I=1,NP
    2     READ(IDEV,200)(B(I,J),J=1,NP)
         FRMEAN=FLOAT(NNOF)/NUTT
```

```
C=> CALCULATE THE OLPC PARAMETERS FOR THE UNKNOWN
C     SPEAKER VIA THE REFERENCE TEMPLATE OF THE REFERENCE
C     SPEAKER NUMBER M.
          DO 5 J=1,NOF
            DO 5 I=1,NP
              DO 5 K=1,NP
    5         Z(I)=Z(I)+B(K,I)*X(K,J)
          DO 6 I=1,NP
    6      Z(I)=Z(I)/NOF
C=> READ IN THE OLPC PARAMETERS FROM THE REFERENCE
C     TEMPLATE OF REFERENCE SPEAKER NUMBER M.
          READ(IDEV,400)(PHI(I),I=1,NP)
  400    FORMAT(/6E15.7/6E15.7)
C=> CALCULATE THE DISTANCE AND STORE IN THE ARRAY 'D'.
          DO 7 I=NPD,NP
    7      D(M)=D(M)+((PHI(I)-Z(I))**2)/LAMDA(I)
    3    D(M)=D(M)*FRMEAN
        RETURN
        END
C**********************************************************
```

Appendix I

DISTANCE MATRICES

TABLE I.1

Statement number: 6 (type A)

Approach: Conventional

NTS ratio: 0

Number of parameters: 12

REFERENCE

TEST	ABBASI	MUFTI	AKHLAQ	JAWED	AHMED	SHABBAR	MATLOOB	TAEMOOR	ZAIDI	RASHEED	HAFEEZ
ABBASI	12.27	183.98	527.79	109.14	432.99	413.29	643.73	523.85	270.30	220.45	678.46
MUFTI	247.81	11.62	614.24	109.69	379.00	659.44	842.09	287.99	351.74	70.27	536.28
AKHLAQ	615.95	390.94	18.62	325.19	333.95	345.51	445.84	330.44	332.22	240.91	220.83
JAWED	338.55	483.18	337.38	65.92	294.53	209.73	167.21	287.16	354.01	323.06	304.91
AHMED	296.63	283.12	1034.30	182.56	18.19	446.53	312.23	154.42	449.26	308.24	485.17
SHABBAR	392.17	1055.66	1155.86	381.43	722.76	11.63	303.93	945.71	907.79	713.23	759.17
MATLOOB	403.77	694.50	503.19	241.57	318.10	148.77	33.99	423.99	553.63	437.65	406.26
TAEMOOR	466.22	149.58	661.29	222.72	282.70	661.38	518.14	9.51	374.83	143.42	384.51
ZAIDI	402.34	206.36	240.83	278.56	456.56	402.38	265.84	373.64	28.28	194.22	468.56
RASHEED	817.96	180.61	674.22	405.40	802.77	969.07	1435.71	360.45	687.54	16.24	732.75
HAFEEZ	1952.84	3089.00	2486.48	1484.56	1922.49	1243.66	1306.06	1853.13	4074.77	1245.08	21.42

TABLE I.2

Statement number: 6 (type A)

Approach: Conventional

NTS ratio: .05

Number of parameters: 12

REFERENCE

TEST	ABBASI	MUFTI	AKHLAQ	JAWED	AHMED	SHABBAR	MATLOOB	TAEMOOR	ZAIDI	RASHEED	HAFEEZ
ABBASI	204.47	239.95	1228.54	159.53	286.99	355.98	510.42	416.54	519.26	281.50	566.24
MUFTI	546.53	219.34	2031.16	299.17	233.89	389.16	474.35	212.89	772.99	329.31	432.23
AKHLAQ	655.67	407.62	24.60	344.56	338.99	347.70	509.41	337.64	391.97	263.05	214.30
JAWED	619.45	505.57	804.65	300.43	211.45	127.85	283.14	285.31	629.23	526.08	267.85
AHMED	1021.71	763.87	2854.14	652.48	380.10	412.22	285.87	471.60	1239.75	763.09	466.03
SHABBAR	499.99	872.98	1463.62	552.54	525.58	79.61	319.51	722.99	945.17	601.73	636.41
MATLOOB	490.79	669.79	955.32	349.33	244.41	157.30	88.07	345.20	746.30	487.34	357.84
TAEMOOR	877.63	392.70	2231.67	450.32	223.91	452.58	329.24	155.60	865.02	466.32	332.56
ZAIDI	405.53	226.38	170.41	246.98	284.06	269.07	174.71	265.94	109.33	203.04	346.24
RASHEED	880.23	270.21	1979.45	378.19	353.57	611.15	854.03	173.34	849.92	152.58	491.63
HAFEEZ	2140.22	2053.90	2639.30	1434.08	832.40	933.64	890.76	1063.08	3301.04	774.77	74.15

TABLE I.3

Statement number: 6 (type A)

Approach: Conventional

NTS ratio: .10

Number of parameters: 12

REFERENCE

TEST	ABBASI	MUFTI	AKHLAQ	JAWED	AHMED	SHABBAR	MATLOOB	TAEMOOR	ZAIDI	RASHEED	HAFEEZ
ABBASI	627.07	414.67	2701.91	438.13	398.07	364.22	464.48	430.16	1019.73	449.63	540.88
MUFTI	1291.44	675.85	4936.46	886.37	660.05	639.20	754.27	523.19	1724.83	626.38	532.15
AKHLAQ	756.82	434.49	87.39	408.34	343.20	377.42	652.56	338.99	524.35	319.21	212.81
JAWED	1214.97	680.14	2081.41	748.36	453.62	179.95	451.89	376.33	1216.09	864.84	338.46
AHMED	2255.53	1549.57	6283.01	1537.98	1106.22	731.82	591.45	1085.18	2674.25	1261.37	637.36
SHABBAR	752.74	957.52	2119.04	737.92	560.45	151.78	351.76	692.24	1257.81	727.14	494.33
MATLOOB	772.95	792.84	1744.37	550.45	374.74	191.97	157.56	418.91	1118.27	634.62	346.20
TAEMOOR	1720.83	919.14	5101.98	1086.81	682.42	649.38	530.13	508.05	1883.28	820.10	420.82
ZAIDI	607.44	317.76	501.64	394.79	298.44	245.16	271.07	252.40	393.32	314.56	297.16
RASHEED	1396.15	689.24	4761.70	858.56	612.87	804.73	1012.29	443.68	1653.32	355.49	482.65
HAFEEZ	2701.23	1950.75	4530.30	1756.27	954.04	900.03	800.03	1064.15	3528.57	874.06	219.53

TABLE I.4

Statement number: 6 (type A)

Approach: AFT

NTS ratio: 0

Number of parameters: 12

REFERENCE

TEST	ABBASI	MUFTI	AKHLAQ	JAWED	AHMED	SHABBAR	MATLOOB	TAEMOOR	ZAIDI	RASHEED	HAFEEZ
ABBASI	12.27	183.98	527.79	109.14	432.99	413.29	643.73	523.85	270.30	220.45	678.46
MUFTI	247.81	11.62	614.24	109.69	379.00	659.44	842.09	287.99	351.74	70.27	536.28
AKHLAQ	611.84	398.55	18.24	321.90	340.25	339.39	444.93	335.37	339.11	243.90	221.90
JAWED	371.91	534.98	374.25	74.26	330.12	224.86	173.77	320.89	404.09	348.52	317.93
AHMED	296.63	283.12	1034.30	182.56	18.19	446.53	312.23	154.42	449.26	308.24	485.17
SHABBAR	392.17	1055.66	1155.86	381.43	722.76	11.63	303.93	945.71	907.79	713.23	759.17
MATLOOB	403.77	694.50	503.19	241.57	318.10	148.77	33.99	423.99	553.63	437.65	406.26
TAEMOOR	466.22	149.58	661.29	222.72	282.70	661.38	518.14	9.51	374.83	143.42	384.51
ZAIDI	402.34	206.36	240.83	278.56	456.56	402.38	265.84	373.64	28.28	194.22	468.56
RASHEED	817.96	180.61	674.22	405.40	802.77	969.07	1435.71	360.45	687.54	16.24	732.75
HAFEEZ	1606.55	2586.02	2023.15	1204.14	1574.20	956.87	1000.85	1511.26	3383.78	1069.24	12.63

TABLE 1.5

Statement number: 6 (type A)

Approach: AFT

NTS ratio: .05

Number of parameters: 12

REFERENCE

TEST	ABBASI	MUFTI	AKHLAQ	JAWED	AHMED	SHABBAR	MATLOOB	TAEMOOR	ZAIDI	RASHEED	HAFEEZ
ABBASI	37.04	183.91	606.27	97.98	259.61	294.84	439.69	409.98	250.65	248.86	679.22
MUFTI	256.77	59.24	967.44	112.53	140.65	512.54	603.35	142.29	438.91	106.33	467.64
AKHLAQ	617.38	376.56	16.89	322.00	341.60	337.55	459.70	332.37	319.21	235.92	226.33
JAWED	229.81	363.66	373.78	80.57	150.73	137.70	106.51	241.19	258.06	315.16	375.07
AHMED	396.39	362.94	1185.20	234.71	55.92	403.31	241.09	199.41	526.33	399.60	465.69
SHABBAR	397.96	830.99	1283.16	454.82	530.79	40.36	277.63	776.05	823.31	557.16	704.95
MATLOOB	321.99	561.41	588.14	236.67	235.43	147.26	39.04	343.19	479.54	408.06	441.93
TAEMOOR	438.86	140.59	1061.73	182.96	127.50	523.78	411.17	30.21	440.16	191.65	386.23
ZAIDI	394.78	195.19	199.75	266.58	403.07	354.98	215.63	312.01	41.48	194.89	427.58
RASHEED	798.51	184.77	1076.58	363.42	489.03	780.25	1134.92	178.38	687.70	56.83	624.10
HAFEEZ	1638.45	1912.57	1615.05	1050.71	780.70	773.38	727.88	941.40	2735.04	744.71	29.12

TABLE I.6

Statement number: 6 (type A)

Approach: AFT

NTS ratio: .10

Number of parameters: 12

REFERENCE

TEST	ABBASI	MUFTI	AKHLAQ	JAWED	AHMED	SHABBAR	MATLOOB	TAEMOOR	ZAIDI	RASHEED	HAFEEZ
ABBASI	113.66	204.00	825.29	108.96	156.79	232.33	330.39	300.20	326.26	308.07	605.73
MUFTI	366.51	135.57	1612.43	197.82	141.64	548.36	655.83	158.49	655.20	167.53	484.61
AKHLAQ	613.30	338.85	19.09	320.08	326.46	334.97	471.90	317.32	280.78	225.81	241.58
JAWED	225.58	299.08	606.12	110.39	86.80	126.52	114.99	186.10	281.39	346.13	390.12
AHMED	559.34	450.76	1666.85	339.33	105.07	450.67	278.49	241.46	731.62	442.27	433.39
SHABBAR	397.24	649.30	1489.86	456.61	427.23	76.05	255.64	613.59	742.08	446.04	602.58
MATLOOB	345.96	494.69	721.71	261.51	227.45	162.68	53.36	309.80	475.16	392.41	442.17
TAEMOOR	524.99	228.58	1714.55	266.22	135.75	516.36	423.10	102.14	647.49	266.21	392.12
ZAIDI	372.29	183.06	190.95	239.91	306.41	320.71	167.97	229.03	65.68	197.45	380.91
RASHEED	875.08	259.50	1833.03	420.95	381.03	805.03	1118.24	169.50	853.76	114.29	583.89
HAFEEZ	1678.74	1679.84	1890.75	1041.17	578.70	701.49	626.66	781.20	2588.72	688.46	63.47

TABLE I.7

Statement number: 6 (type A)

Approach: ANC

NTS ratio: 0

Number of parameters: 12

REFERENCE

TEST	ABBASI	MUFTI	AKHLAQ	JAWED	AHMED	SHABBAR	MATLOOB	TAEMOOR	ZAIDI	RASHEED	HAFEEZ
ABBASI	153.84	110.62	740.21	107.40	318.96	389.08	604.61	235.50	506.43	194.56	600.10
MUFTI	324.74	75.51	1080.43	236.21	285.63	685.54	1024.65	240.58	669.19	119.19	633.16
AKHLAQ	444.01	198.56	190.80	220.94	333.19	353.03	441.03	200.98	158.06	153.51	298.16
JAWED	164.19	184.53	368.50	50.90	136.23	196.84	240.54	166.42	201.84	183.02	383.80
AHMED	304.69	211.65	1048.17	225.06	103.93	375.97	415.64	195.41	578.38	296.38	478.98
SHABBAR	231.85	579.87	913.64	254.10	496.64	57.37	264.42	648.29	601.62	487.75	697.89
MATLOOB	192.70	318.26	612.33	169.52	260.59	201.70	78.83	272.64	271.50	327.70	594.26
TAEMOOR	481.07	145.96	752.33	368.12	592.35	530.82	742.72	226.45	512.63	168.07	688.58
ZAIDI	300.67	91.29	314.35	205.45	285.90	323.49	302.54	179.98	156.91	141.76	460.11
RASHEED	719.83	167.51	780.45	430.95	732.83	848.88	1324.70	188.46	736.20	95.29	635.46
HAFEEZ	612.34	816.83	1037.55	460.51	493.18	593.11	672.69	403.11	1320.83	359.40	96.72

TABLE I.8

Statement number: 6 (type A)

Approach: ANC

NTS ratio: .05

Number of parameters: 12

REFERENCE

TEST	ABBASI	MUFTI	AKHLAQ	JAWED	AHMED	SHABBAR	MATLOOB	TAEMOOR	ZAIDI	RASHEED	HAFEEZ
ABBASI	248.77	150.19	838.09	161.25	206.67	385.19	505.21	214.29	489.57	282.68	640.45
MUFTI	619.43	245.15	2251.37	341.79	301.61	561.38	758.51	323.23	922.40	310.67	588.83
AKHLAQ	437.70	214.70	155.39	215.58	317.05	334.24	456.66	203.65	173.20	160.76	299.55
JAWED	239.19	333.98	755.74	153.95	196.96	174.52	260.83	261.97	386.18	439.07	469.06
AHMED	457.44	257.03	1664.39	286.65	151.36	370.08	369.65	281.66	706.62	374.77	528.29
SHABBAR	225.31	469.07	1138.27	313.37	312.45	80.18	236.39	488.50	543.40	406.62	697.54
MATLOOB	198.34	281.94	824.66	200.97	187.70	212.06	101.04	229.50	297.62	318.42	557.00
TAEMOOR	481.19	245.78	1357.11	333.41	361.06	358.50	476.00	257.48	609.39	365.88	633.22
ZAIDI	284.51	92.01	374.62	175.89	227.39	281.72	281.12	185.49	192.05	148.09	466.72
RASHEED	513.23	123.41	1422.94	266.25	304.16	681.88	991.85	122.30	672.09	83.93	572.66
HAFEEZ	586.05	631.25	1221.29	359.45	273.49	506.31	496.51	274.23	1022.10	297.69	156.82

TABLE I.9

Statement number: 6 (type A)

Approach: ANC

NTS ratio: .10

Number of parameters: 12

REFERENCE

TEST	ABBASI	MUFTI	AKHLAQ	JAWED	AHMED	SHABBAR	MATLOOB	TAEMOOR	ZAIDI	RASHEED	HAFEEZ
ABBASI	1509.43	772.08	4907.57	1002.19	577.42	943.54	978.21	420.65	1750.76	541.31	414.51
MUFTI	1084.77	504.01	4327.29	679.62	572.34	892.26	1211.12	471.12	1650.32	522.94	712.52
AKHLAQ	416.15	241.22	86.48	198.83	279.71	301.30	459.97	201.21	184.40	176.80	267.65
JAWED	919.79	465.48	1825.92	535.33	329.12	299.92	520.76	265.47	922.58	807.27	403.97
AHMED	791.34	489.36	2869.67	547.93	272.57	550.10	572.80	402.26	1107.44	391.25	505.25
SHABBAR	215.16	429.41	964.16	291.06	223.65	111.50	239.48	336.95	507.96	411.62	538.37
MATLOOB	225.85	267.82	1005.30	236.53	170.35	233.61	136.41	199.73	338.47	313.91	510.64
TAEMOOR	758.26	511.78	2472.22	534.59	347.89	586.64	522.27	359.06	991.42	571.10	621.56
ZAIDI	204.09	116.10	492.61	151.62	150.53	199.81	206.91	204.06	214.13	190.53	459.08
RASHEED	844.14	271.75	2664.87	428.83	368.44	809.28	1112.72	202.40	1087.84	214.16	536.84
HAFEEZ	765.35	790.31	1395.34	422.02	204.04	410.36	344.70	309.84	1144.62	418.73	166.57

TABLE I.10

Statement number: 6 (type A)

Approach: IV

NTS ratio: .05

Number of parameters: 12

REFERENCE

TEST	ABBASI	MUFTI	AKHLAQ	JAWED	AHMED	SHABBAR	MATLOOB	TAEMOOR	ZAIDI	RASHEED	HAFEEZ
ABBASI	0.17E 06	0.16E 06	0.11E 06	0.14E 06	0.20E 06	0.87E 05	0.14E 06	0.21E 06	0.22E 06	0.14E 06	0.14E 06
MUFTI	0.27E 04	0.41E 04	0.56E 04	0.32E 04	0.46E 04	0.37E 04	0.72E 04	0.79E 04	0.49E 04	0.26E 04	0.54E 04
AKHLAQ	0.13E 05	0.11E 05	0.62E 04	0.11E 05	0.12E 05	0.61E 04	0.88E 04	0.10E 05	0.12E 05	0.62E 04	0.61E 04
JAWED	0.25E 05	0.25E 05	0.15E 05	0.18E 05	0.21E 05	0.71E 04	0.20E 05	0.20E 05	0.20E 05	0.23E 05	0.13E 05
AHMED	0.20E 09	0.16E 09	0.30E 09	0.16E 09	0.22E 09	0.13E 09	0.22E 09	0.31E 09	0.19E 09	0.13E 09	0.23E 09
SHABBAR	0.12E 06	0.11E 06	0.11E 06	0.10E 06	0.14E 06	0.66E 05	0.99E 05	0.14E 06	0.13E 06	0.63E 05	0.90E 05
MATLOOB	0.19E 07	0.19E 07	0.21E 07	0.13E 07	0.25E 07	0.11E 07	0.10E 07	0.20E 07	0.28E 07	0.14E 07	0.79E 06
TAEMOOR	0.47E 04	0.60E 04	0.42E 04	0.38E 04	0.42E 04	0.43E 04	0.55E 04	0.41E 04	0.88E 04	0.29E 04	0.23E 04
ZAIDI	0.13E 04	0.16E 04	0.18E 04	0.12E 04	0.10E 04	0.13E 04	0.74E 03	0.13E 04	0.15E 04	0.12E 04	0.14E 04
RASHEED	0.14E 04	0.12E 04	0.23E 04	0.14E 04	0.16E 04	0.11E 04	0.99E 03	0.11E 04	0.14E 04	0.85E 03	0.93E 03
HAFEEZ	0.24E 06	0.24E 06	0.33E 06	0.25E 06	0.23E 06	0.24E 06	0.25E 06	0.32E 06	0.24E 06	0.11E 06	0.30E 06

TABLE I.11

Statement number: 6 (type A)

Approach: AS

NTS ratio: .05

Number of parameters: 12

	REFERENCE										
TEST	ABBASI	MUFTI	AKHLAQ	JAWED	AHMED	SHABBAR	MATLOOB	TAEMOOR	ZAIDI	RASHEED	HAFEEZ
ABBASI	10103.98	7626.90	7455.90	7124.92	8081.98	7045.59	7441.77	7833.51	10271.66	4769.10	7008.21
MUFTI	3259.71	3002.79	3803.15	1934.97	3427.57	1435.83	1569.16	2917.32	2632.84	2417.21	1590.51
AKHLAQ	625.27	383.84	18.82	320.41	342.93	339.99	457.77	336.15	326.15	236.41	221.21
JAWED	16131.88	15025.61	9528.20	11627.95	10194.85	9327.91	8768.17	12464.06	18319.75	6819.82	6742.95
AHMED	2511.52	2624.31	1877.51	2443.59	2751.64	1603.16	1738.17	2735.37	2242.33	1946.61	2482.32
SHABBAR	16102.13	16299.93	17925.81	12424.18	10645.17	6570.54	4872.07	10926.32	22137.41	10829.65	12176.04
MATLOOB	819.51	867.12	1399.88	611.56	905.44	363.25	374.69	802.79	986.63	615.96	784.64
TAEMOOR	8525.14	11839.39	8252.81	9519.40	10576.75	2767.04	5282.38	10635.44	9594.23	7184.95	5855.74
ZAIDI	518.29	379.45	735.39	574.99	1082.90	747.12	861.27	1200.95	255.04	275.34	1105.83
RASHEED	4282.56	3722.09	4134.92	2613.61	3358.40	1290.68	2790.67	3506.91	5218.62	3252.88	2556.02
HAFEEZ	11906.91	11025.30	11980.36	13051.95	14486.04	12358.95	10729.82	12981.44	19526.74	7768.95	3490.87

TABLE I.12

Statement number: 6 (type A)

Approach: SYW

NTS ratio: .05

Number of parameters: 12

REFERENCE

TEST	ABBASI	MUFTI	AKHLAQ	JAWED	AHMED	SHABBAR	MATLOOB	TAEMOOR	ZAIDI	RASHEED	HAFEEZ
ABBASI	0.58E 06	0.38E 06	0.34E 06	0.43E 06	0.35E 06	0.29E 06	0.46E 06	0.40E 06	0.50E 06	0.26E 06	0.33E 06
MUFTI	0.44E 06	0.52E 06	0.40E 06	0.37E 06	0.27E 06	0.22E 06	0.24E 06	0.29E 06	0.56E 06	0.52E 06	0.18E 06
AKHLAQ	0.12E 05	0.70E 04	0.78E 04	0.95E 04	0.87E 04	0.62E 04	0.12E 05	0.83E 04	0.12E 05	0.37E 04	0.38E 04
JAWED	0.84E 06	0.65E 06	0.49E 06	0.53E 06	0.45E 06	0.28E 06	0.37E 06	0.52E 06	0.98E 06	0.38E 06	0.32E 06
AHMED	0.26E 08	0.25E 08	0.23E 08	0.23E 08	0.20E 08	0.11E 08	0.19E 08	0.20E 08	0.26E 08	0.18E 08	0.12E 08
SHABBAR	0.16E 05	0.11E 05	0.15E 05	0.13E 05	0.12E 05	0.50E 04	0.87E 04	0.13E 05	0.14E 05	0.65E 04	0.95E 04
MATLOOB	0.17E 05	0.21E 05	0.18E 05	0.16E 05	0.14E 05	0.13E 05	0.11E 05	0.18E 05	0.22E 05	0.11E 05	0.12E 05
TAEMOOR	0.11E 06	0.12E 06	0.39E 05	0.85E 05	0.98E 05	0.47E 05	0.14E 06	0.69E 05	0.11E 06	0.13E 06	0.47E 05
ZAIDI	0.13E 05	0.13E 05	0.32E 05	0.13E 05	0.12E 05	0.14E 05	0.22E 05	0.17E 05	0.23E 05	0.85E 04	0.14E 05
RASHEED	0.10E 05	0.13E 05	0.18E 05	0.86E 04	0.15E 05	0.87E 04	0.11E 05	0.11E 05	0.12E 05	0.75E 04	0.61E 04
HAFEEZ	0.60E 05	0.97E 05	0.10E 06	0.43E 05	0.10E 06	0.36E 05	0.25E 05	0.61E 05	0.79E 05	0.45E 05	0.18E 05

TABLE I.13

Statement number: 6 (type A)

Approach: LPS

NTS ratio: .05

Number of parameters: 12

REFERENCE

TEST	ABBASI	MUFTI	AKHLAQ	JAWED	AHMED	SHABBAR	MATLOOB	TAEMOOR	ZAIDI	RASHEED	HAFEEZ
ABBASI	579.04	607.14	3163.31	636.27	496.48	464.50	571.59	581.37	1117.33	507.38	865.06
MUFTI	1133.23	776.55	5609.20	1058.48	820.42	1028.65	1368.79	712.05	1857.30	435.68	691.14
AKHLAQ	623.72	431.44	616.55	577.44	579.09	394.95	532.89	483.58	389.18	375.86	555.68
JAWED	521.26	563.74	2460.25	551.55	454.05	432.67	328.09	481.82	827.31	558.10	710.61
AHMED	956.70	783.82	3997.09	906.04	725.97	645.02	638.85	737.34	1469.55	624.67	615.26
SHABBAR	1440.37	999.59	6639.85	1339.48	1123.71	1065.43	1552.93	773.22	2023.50	484.55	823.75
MATLOOB	700.66	590.38	2808.98	665.60	593.81	479.05	433.78	454.37	907.01	464.97	688.19
TAEMOOR	1370.82	910.20	6280.10	1187.10	1020.32	904.94	1054.16	830.66	1995.44	661.33	729.24
ZAIDI	771.60	553.39	2318.21	734.08	596.77	568.08	596.59	445.11	743.49	427.44	618.35
RASHEED	1468.90	848.62	7290.65	1160.86	1057.97	1514.95	2317.48	710.94	2368.46	481.49	853.46
HAFEEZ	3571.81	2893.01	10998.91	2889.32	2356.72	1878.97	2351.61	1727.84	5258.84	1001.25	612.22